



TUGAS AKHIR - KI141502

RANCANG BANGUN APLIKASI PERMAINAN SIMULASI BERTAHAN HIDUP DI SEBUAH PULAU TERPENCIL

Muhammad Wildianurahman

NRP 5112 100 165

Dosen Pembimbing I

Imam Kuswardayan, S.Kom., M.T

Dosen Pembimbing II

Dr. Eng. Darlis Herumurti, S.Kom., M.Kom.

DEPARTEMEN INFORMATIKA

Fakultas Teknologi Informasi & Komunikasi

Institut Teknologi Sepuluh Nopember

Surabaya 2018



TUGAS AKHIR - KI141502

RANCANG BANGUN APLIKASI PERMAINAN SIMULASI BERTAHAN HIDUP DI SEBUAH PULAU TERPENCIL

Muhammad Wildianurahman

NRP 5112 100 165

Dosen Pembimbing I

Imam Kuswardayan, S.Kom., M.T.

Dosen Pembimbing II

Dr. Eng. Darlis Herumurti, S.Kom., M.Kom.

DEPARTEMEN INFORMATIKA

Fakultas Teknologi Informasi & Komunikasi

Institut Teknologi Sepuluh Nopember

Surabaya 2018

(Halaman Ini Sengaja Dikosongkan)



FINAL PROJECT - KI141502

DESIGN OF SURVIVAL IN DESERTED ISLAND SIMULATION GAME

Muhammad Wildianurahman

NRP 5112 100 165

Supervisor I

Imam Kuswardayan, S.Kom., M.T.

Supervisor II

Dr. Eng. Darlis Herumurti, S.Kom., M.Kom.

DEPARTMENT OF INFORMATICS

Faculty of Information Technology & Communication

Institut Teknologi Sepuluh Nopember

Surabaya 2018

(Halaman Ini Sengaja Dikosongkan)

LEMBAR PENGESAHAN

RANCANG BANGUN APLIKASI PERMAINAN SIMULASI BERTAHAN HIDUP DI SEBUAH PULAU TERPENCIL

Tugas Akhir

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Bidang Studi Mata Kuliah Interaksi, Grafika, dan Seni
Program Studi S-1 Departemen Informatika
Fakultas Teknologi Informasi & Komunikasi
Institut Teknologi Sepuluh Nopember

Oleh:

MUHAMMAD WILDIANURAHMAN

NRP. 5112 100 165

Disetujui oleh Dosen Pembimbing Tugas Akhir:

Imam Kuswardayan, S.Kom., M.T.

NIP: 19761215 200312 1 001

Dr. Eng. Darlis Herumurti, S.Kom., M.Kom.

NIP: 19771217 200312 1 001



**SURABAYA
JANUARI, 2018**

RANCANG BANGUN APLIKASI PERMAINAN SIMULASI BERTAHAN HIDUP DI SEBUAH PULAU TERPENCIL

Nama Mahasiswa : Muhammad Wildianurahman
NRP : 5112 100 165
Jurusan : Departemen Informatika - FTIK-ITS
Dosen Pembimbing I : Imam Kuswardayan, S.Kom., M.T.
Dosen Pembimbing II : Dr. Eng. Darlis Herumurti, S.Kom.,
M.Kom.

ABSTRAK

Sekitar ratusan game bergenre simulasi dirilis setiap tahunnya. Akan tetapi, diantara semua game simulasi kehidupan yang telah rilis, game simulasi kehidupan yang juga memiliki genre survival jumlahnya tidak terlalu banyak. Untuk mengatasi kekurangan itu, maka penulis membuat aplikasi game simulasi bertahan hidup

Pada tugas akhir ini penulis membuat aplikasi simulasi bertahan hidup di pulau terpencil dengan mengimplementasikan finite state machine dan faktor-faktor bertahan hidup yang kebanyakan digunakan di dalam game bergenre survival. Aplikasi simulasi bertahan hidup di pulau terpencil ini dapat digunakan pada perangkat bergerak dengan sistem operasi android. Aplikasi dibangun menggunakan game engine Unity 5.5.2f1 (64-bit) dan Android SDK 23.0.1.

Uji coba pada tugas akhir ini menggunakan skenario uji coba dengan metode pengambilan kuesioner dari pengguna. Penguji aplikasi tugas akhir ini adalah 33 orang yang secara acak ditemukan penulis. Dari hasil kuesioner yang telah diajukan kepada penguji, aplikasi ini menunjukkan bahwa faktor-faktor bertahan hidup yang diimplementasikan mempengaruhi tindakan

pengguna karena rekapitulasi data dari kuesioner menunjukkan bahwa 73.98% dari seluruh pertanyaan tentang faktor-faktor bertahan hidup dijawab setuju oleh penguji.

Kata kunci: Simulasi, bertahan hidup, pulau terpencil, android, psikologi.

DESIGN OF SURVIVAL IN DESERTED ISLAND SIMULATION GAME

Student Name : Muhammad Wildianurahman
NRP : 5112 100 165
Major : Informatics Department - FTIK-ITS
Advisor I : Imam Kuswardayan, S.Kom., M.T.
Advisor II : Dr. Eng. Darlis Herumurti, S.Kom.,
M.Kom..

ABSTRACT

About more than one hundred games with simulation genre released every year. But, between all life simulation game that already released, there isn't much life simulation game that also have survival genre. To overcome this shortage, writer make survival simulation game application.

In this final project, the writer make simulation game application of survival in deserted island by implementing the factors of survival that mostly used in survival genre game. simulation game application of survival in deserted island can be used on mobile devices with android operating system. The application is built using the game engine Unity 5.5.2f1 (64-bit) and Android SDK 23.0.1.

Trial on this final project using trial scenario with method of making questioner from user. Tester of this final application is 33 people who are randomly found by writer. From the results of the questionnaires that have been submitted to the examiners, this application shows that the factors of survival that implemented influence the user's action because the data recapitulation of the questionnaire shows 73.98% of all questions about survival answered agreed by testers.

Keywords: *Simulation, survival, deserted island, android, psychology.*

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Segala puji bagi Allah SWT, Tuhan semesta alam yang telah melimpahkan rahmat dan hidayah-Nya kepada penulis, sehingga tugas akhir berjudul “Rancang Bangun Aplikasi Permainan Simulasi Bertahan Hidup Di Sebuah Pulau Terpencil” ini dapat selesai sesuai dengan waktu yang telah ditentukan. Pengerjaan tugas akhir ini menjadi sebuah sarana untuk penulis memperdalam ilmu yang telah didapatkan selama menempuh pendidikan di kampus perjuangan Institut Teknologi Sepuluh Nopember Surabaya, khususnya dalam disiplin ilmu Teknik Informatika. terselesaikannya buku tugas akhir ini tidak terlepas dari bantuan dan dukungan semua pihak. Penulis ingin mengucapkan terima kasih kepada:

1. Allah SWT atas limpahan rahmat dan rezeki-Nya sehingga penulis dapat menyelesaikan Tugas Akhir dan mendapat ilmu dalam waktu pengerjaannya.
2. Kedua orang tua penulis yang telah memberikan doa, dukungan moral dan material yang tak pernah putus demi kesuksesan penulis.
3. Bapak Imam Kuswardayan selaku dosen pembimbing 1 dan juga dosen wali penulis yang telah membimbing penulis dalam menyelesaikan tugas akhir ini dan memberikan saran dan motivasi pada penulis.

4. Bapak Darlis Herumurti selaku dosen pembimbing 2 yang telah membimbing penulis dalam menyelesaikan tugas akhir ini.
5. Teman - teman, yang berlalu-lalang menemani penulis selama kuliah dengan ingatan, pelajaran, dan canda tawa.
6. Sahabat-sahabat penulis yang secara pribadi menjadi motivasi dan wadah saling belajar bagi penulis.
7. Bapak Radityo Anggoro selaku dosen koordinator Tugas Akhir yang telah membantu penulis dalam segala sesuatu terikat peraturan akademis yang memperlancar terlaksananya proses pembuatan sampai sidang Tugas Akhir.
8. Segenap dosen-dosen dan karyawan Teknik Informatika yang dengan sabar mendidik dan memberikan pengalaman baru kepada penulis selama di Teknik Informatika.
9. Semua pihak yang sudah membantu dan tidak bisa penulis tuliskan satu persatu.

Penulis menyadari bahwa tugas akhir ini masih memiliki banyak kekurangan. Dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depan. Akhir kata, penulis berharap semoga kita semua berhasil sukses mendapatkan kebaikan dunia akhirat.

Surabaya, Desember 2017

Muhammad Wildianurahman

DAFTAR ISI

LEMBAR PENGESAHAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
DAFTAR KODE SUMBER	xix
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan	2
1.3 Rumusan Masalah	2
1.4 Batasan Masalah	2
1.5 Manfaat	3
1.6 Metodologi	3
1.7 Sistematika Penulisan	4
BAB II TINJAUAN PUSTAKA	7
2.1 Simulasi game	7
2.2 Life simulation game	8
2.3 Survival game	8
2.4 Finite State Machine	9
2.5 Unity	10
2.6 Android SDK	11
2.7 60 Seconds	11
BAB III ANALISIS DAN PERANCANGAN	12
3.1 Analisis Sistem	12
3.2 Perancangan Aplikasi	13
3.2.1 Deskripsi Umum Perangkat Lunak	13
3.2.2 Rancangan Penentuan Untuk Menang Dan Kalah	13
3.2.3 Rancangan Faktor-Faktor Yang Terlibat Dalam Simulasi Bertahan Hidup	14
3.2.3.1 Status Pemain	14

3.2.3.2	Tempat Yang Bisa Dikunjungi Di Pulau Terencil.....	29
3.2.3.3	Waktu	34
3.2.3.4	Kejadian.....	34
3.2.4	Perancangan Skenario Aplikasi	51
3.3	Perancangan Tampilan Antarmuka	53
3.3.1	Tampilan Menu Utama.....	53
3.3.2	Tampilan Bagian Awal.....	54
3.3.3	Tampilan Menu Pause	55
3.3.4	Tampilan Menu Inventory	56
3.3.5	Tampilan Bagian Makan Dan Minum	58
3.3.6	Tampilan Bagian Event	59
BAB IV IMPLEMENTASI.....		63
4.1	Lingkungan Implementasi	63
4.2	Implementasi Antarmuka.....	63
4.3	Implementasi Faktor-Faktor Bertahan Hidup dan Finite State Machine.....	67
4.3.1	Penentuan Untuk Menang dan Kalah	67
4.3.2	Status.....	69
4.3.3	Tempat	80
4.3.4	Waktu.....	87
4.3.5	Kejadian	91
BAB V PENGUJIAN DAN EVALUASI.....		105
5.1	Lingkungan Uji Coba	105
5.2	Pengujian Aturan Menang Dan Kalah Dan Faktor-Faktor Bertahan Hidup	105
5.3	Pengujian Pengguna.....	113
5.4	Evaluasi	115
BAB VI KESIMPULAN DAN SARAN		123
6.1.	Kesimpulan.....	123
6.2.	Saran.....	124
BIODATA PENULIS		127

DAFTAR GAMBAR

Gambar 3.1 Finite state machine hungry dan thirsty	17
Gambar 3.2 Finite state machine life	27
Gambar 3.3 Finite state machine sanity	28
Gambar 3.4 Finite state machine tempat	29
Gambar 3.5 Finite state machine pilihan dengan barang	51
Gambar 3.6 Rancangan tampilan menu utama	54
Gambar 3.7 Rancangan antarmuka bagian awal	55
Gambar 3.8 Rancangan antarmuka menu pause	56
Gambar 3.9 Rancangan antarmuka menu inventory	57
Gambar 3.10 Rancangan antarmuka bagian makan dan minum	59
Gambar 3.11 Rancangan antarmuka bagian event	60
Gambar 3.12 Rancangan event bar	61
Gambar 4.1 Antarmuka menu utama	63
Gambar 4.2 Antarmuka bagian awal	64
Gambar 4.3 Antarmuka menu pause	64
Gambar 4.4 Antarmuka menu inventory	65
Gambar 4.5 Antarmuka bagian makan dan minum	65
Gambar 4.6 Antarmuka bagian event	66
Gambar 4.7 Antarmuka event bar	66
Gambar 4.8 Scene pemain menang	67
Gambar 4.9 Scene pemain kalah	68
Gambar 4.10 Status saat memulai permainan	69
Gambar 4.11 Kondisi awal makan dan minum	74
Gambar 4.12 Kondisi pemain memilih makan dan minum	75
Gambar 4.13 Status hungry dan thirsty setelah memilih tidak makan dan tidak minum	77
Gambar 4.14 Status hungry dan thirsty setelah memilih tidak makan dan tidak minum (2)	77
Gambar 4.15 Pemain mulai dari home	81
Gambar 4.16 Pemain berada di sungai	82
Gambar 4.17 Pemain berada di gua	83
Gambar 4.18 Pemain berada di home kembali waktu masuk ke hari berikutnya	83

Gambar 4.19 Menerima event dari tempat yang dipilih	92
Gambar 4.20 Contoh event menambah atau mengurangi status .	96
Gambar 4.21 Contoh event biasa tanpa pilihan	96
Gambar 4.22 Contoh event mendapatkan barang	98
Gambar 4.23 Barang di menu inventory setelah mendapatkan panah	101
Gambar 4.24 Contoh event yang memerlukan barang untuk mengakses salah satu pilihannya	102

DAFTAR TABEL

Tabel 3.1Tingkatan status hungry dan thirsty	15
Tabel 3.2 Penjelasan finite state machine hungry dan thirsty	18
Tabel 3.3 Tingkatan status life	24
Tabel 3.4 Tingkatan status sanity	25
Tabel 3.5 Penjelasan finite state machine tempat	29
Tabel 3.6 Contoh event dan efek pilihannya	35
Tabel 5.1 Uji coba aturan menang dan kalah.....	105
Tabel 5.2 Uji coba status hungry dan thirsty	107
Tabel 5.3 Uji coba status life	109
Tabel 5.4 Uji coba status sanity	109
Tabel 5.5 Uji coba event.....	110
Tabel 5.6 Menemukan keberadaan faktor-faktor bertahan hidup	112
Tabel 5.7 Menemukan implementasi finite state machine di dalam permainan.....	112
Tabel 5.8 Pertanyaan-pertanyaan kuesioner	113
Tabel 5.9 Rekapitulasi jawaban kuesioner berdasar pertanyaan1 16	
Tabel 5.10 Rekapitulasi jawaban kuesioner berdasar nilai	119
Tabel 5.11 Rekapitulasi jawaban tempat yang sering dikunjungi	119
Tabel 5.12 Rekapitulasi jawaban waktu makan dan minum	120
Tabel 5.13 Rekapitulasi jawaban memenangkan permainan	121
Tabel 5.14 Rekapitulasi jawaban jumlah hari bertahan hidup jika tidak memenangkan permainan	121

(Halaman Ini Sengaja Dikosongkan)

DAFTAR KODE SUMBER

Kode Sumber 4.1 Penentuan menang dan kalah.....	68
Kode Sumber 4.2 Mengatur status	70
Kode Sumber 4.3 Mengatur status saat memulai permainan	70
Kode Sumber 4.4 Status hungry dan thirsty	71
Kode Sumber 4.5 Status life.....	72
Kode Sumber 4.6 Status sanity.....	73
Kode Sumber 4.7 Memilih makan.....	75
Kode Sumber 4.8 Memilih minum	76
Kode Sumber 4.9 Pengecekan makan	78
Kode Sumber 4.10 Pengecekan minum.....	79
Kode Sumber 4.11 Contoh input life dan sanity.....	80
Kode Sumber 4.12 Mengatur tempat berada	81
Kode Sumber 4.13 Mengatur tempat berada saat mulai	81
Kode Sumber 4.14 Membaca objek panah biru di masing-masing tempat dan pemasangan awal	84
Kode Sumber 4.15 Menampilkan tempat pemain berada	85
Kode Sumber 4.16 Input tempat pemain berada dan kembali ke home di hari berikutnya.....	86
Kode Sumber 4.17 Mengatur waktu dan hari	87
Kode Sumber 4.18 Mengatur waktu dan hari saat mulai permainan	87
Kode Sumber 4.19 Menampilkan hari di bagian event.....	88
Kode Sumber 4.20 Menampilkan waktu dan hari di event	88
Kode Sumber 4.21 Mengecek Waktu	90
Kode Sumber 4.22 Input mengganti waktu dan hari	90
Kode Sumber 4.23 Pengaturan awal event bar	91
Kode Sumber 4.24 Mengakses kumpulan event gua dan menampilkan event yang dipilih.....	94
Kode Sumber 4.25 Menyusun kumpulan event-event gua	94
Kode Sumber 4.26 Menyusun dan mengakses kumpulan event dari semua tempat saat memulai permainan	95
Kode Sumber 4.27 Contoh event dengan efek pilihan menambah atau mengurangi status	97

Kode Sumber 4.28 Contoh event biasa tanpa pilihan	98
Kode Sumber 4.29 Contoh event mendapatkan barang	99
Kode Sumber 4.30 Menampilkan barang di menu inventory ...	101
Kode Sumber 4.31 Contoh event yang memerlukan barang untuk mengakses salah satu pilihannya	103

BAB I

PENDAHULUAN

1.1 Latar Belakang

Sekitar ratusan game bergenre simulasi dirilis setiap tahunnya, game-game simulasi yang dirilis ada berbagai jenis. Diantara semua game simulasi yang dirilis, subgenre game simulasi yang populer adalah simulasi konstruksi dan manajemen, simulasi olahraga, dan simulasi kehidupan. Sebagai salah satu subgenre yang populer di game simulasi, game simulasi kehidupan memiliki banyak contoh sebagai game simulasi, contohnya seperti game-game *digital pet*, game simulasi kehidupan suatu hewan (contohnya *Lion* dan *Wolf*), game simulasi sosial (contohnya *The Sims*), game simulasi *dating*, dan lain-lain. Akan tetapi, diantara semua game simulasi kehidupan yang telah rilis, game simulasi kehidupan yang juga memiliki genre *survival* jumlahnya tidak terlalu banyak [1][2].

Untuk mengatasi kekurangan itu, maka penulis akan merancang aplikasi game simulasi bertahan hidup yang dimana perancangannya menggunakan *finite state machine* dan *gameplay* nya berdasarkan *random event-based*. Dengan menggunakan *gameplay* yang berdasarkan *random event-based*, pemain akan dihadapkan dengan *event-event* yang dipilih secara acak dan pilihan-pilihan yang harus dipilih di setiap *event*. Pilihan yang dipilih pemain akan menentukan kemana arah permainan seorang pemain akan berjalan.

Dalam tugas akhir ini, dengan menggunakan *gameplay random event-based*, *event* yang diterima setiap pemain nanti akan bervariasi dan membuat *event-event* yang diterima di hari yang sama antara satu pemain dengan pemain lain, dan juga antara satu sesi game dengan sesi game yang lain setiap pemain berbeda-beda. Dengan gitu diharapkan pemain paling tidak bisa bertahan hidup selama 30 hari dalam kondisi tersebut dan mengetahui apa yang paling dibutuhkan

dan harus dilakukan oleh pemain untuk bisa bertahan hidup di pulau terpencil di dalam *game* ini.

1.2 Tujuan

Tujuan dari pembuatan tugas akhir ini adalah sebagai berikut:

1. Membuat aplikasi permainan berkategori simulasi yang mengimplementasikan faktor survival.
2. Mempelajari tingkah laku seseorang jika dihadapkan pada kondisi tertentu
3. Mengetahui seberapa lama seseorang dapat bertahan hidup di dalam permainan simulasi bertahan hidup ini.

1.3 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana rancangan aturan permainan pada permainan simulasi bertahan hidup di pulau terpencil?
2. Bagaimana menentukan faktor-faktor yang terlibat dalam rancangan simulasi permainan bertahan hidup?
3. Bagaimana merancang pemodelan finite-state machine pada aplikasi permainan simulasi bertahan hidup?
4. Bagaimana mengimplementasikan permainan simulasi bertahan hidup pada perangkat mobile?

1.4 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan antara lain :

1. Aplikasi dibangun menggunakan game engine Unity versi 5.5.2f1 (64-bit) dan Android SDK 23.0.1.
2. Aplikasi dapat digunakan pada perangkat bergerak.
3. Tampilan di dalam aplikasi adalah 2D

1.5 Manfaat

Manfaat dari hasil pembuatan tugas akhir ini antara lain :

1. Pemain dapat menguji penalaran yang dimilikinya.
2. Pemain belajar untuk beradaptasi dengan lingkungan dalam game

1.6 Metodologi

1. Penyusunan proposal tugas akhir.

Proposal tugas akhir ini berisi tentang deskripsi pendahuluan dari tugas akhir yang akan dibuat. Pendahuluan ini terdiri dari hal yang menjadi latar belakang diajukan nya usulan tugas akhir, rumusan masalah yang diangkat, batasan masalah untuk tugas akhir, tujuan dari pembuatan tugas akhir serta manfaat dari hasil pembuatan tugas akhir. Dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir. Subbab metodologi berisi penjelasan mengenai tahapan penyusunan tugas akhir mulai dari penyusunan proposal hingga penyusunan buku tugas akhir. Terdapat pula sub bab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir

2. Studi literatur

Pada studi literatur ini, akan dipelajari *Finite State Machine* sebagai metode yang digunakan di dalam *game* ini. Selain itu, dipelajari juga hal-hal yang berhubungan dengan faktor-faktor untuk bertahan hidup.

3. Analisis dan desain perangkat lunak simulasi bertahan hidup di pulau terpencil

Fitur yang akan terdapat pada aplikasi ini diantaranya adalah:

1. Satu orang pengguna.
2. Grafik 2 dimensi.
3. *Random event based*.
4. 4 status yang menentukan kondisi pemain di permainan

4. Implementasi perangkat lunak

Aplikasi yang akan dibangun adalah aplikasi permainan simulasi yang dapat berjalan di perangkat Android. Dalam membangun aplikasi ini, penulis menggunakan *game engine* Unity. Untuk pengkodean *script* pada aplikasi yang dibuat, penulis menggunakan Visual Studio 2015.

5. Pengujian dan evaluasi

Pengujian aplikasi ini akan dilakukan dengan pengambilan data melalui kuesioner. Pengujian ini dilakukan untuk mengetahui apakah pemain bisa memenangkan *game* simulasi bertahan hidup atau tidak, seberapa lama pemain dapat bertahan hidup, dan juga mengetahui tindakan-tindakan yang diambil pemain saat memainkan *game* simulasi bertahan hidup ini.

6. Penyusunan buku tugas akhir

Pada tahapan ini disusun buku yang memuat dokumentasi mengenai pembuatan serta hasil dari implementasi perangkat lunak yang telah dibuat.

1.7 Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini:

Bab I Pendahuluan

Bab ini berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan tugas akhir. Selain itu, rumusan permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Tinjauan Pustaka

Bab ini berisi penjelasan secara detil mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan tugas akhir ini.

Bab III Analisis dan Perancangan

Bab ini membahas tahap analisis permasalahan dan perancangan dari sistem yang akan dibangun. Analisis permasalahan membahas permasalahan yang diangkat dalam pengerjaan tugas akhir. Perancangan yang dibuat adalah perancangan *finite state machine*, perancangan status pemain, barang, dan tempat-tempat di dalam game, dan perancangan *event-event* dan efek dari setiap pilihan dari suatu *event*. Dijelaskan pula aturan untuk menang dan kalah di dalam game simulasi bertahan hidup.

Bab IV Implementasi

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Bab ini berisi proses implementasi dari setiap tampilan game simulasi bertahan hidup yang dibuat.

Bab V Pengujian Dan Evaluasi

Bab ini membahas pengujian dengan pengambilan data melalui kuesioner. Pengujian ini dilakukan untuk mengetahui apakah pemain bisa memenangkan game simulasi bertahan hidup atau tidak, seberapa lama pemain dapat bertahan hidup, dan juga mengetahui tindakan-tindakan yang diambil pemain saat memainkan game simulasi bertahan hidup ini..

Bab VI Kesimpulan Dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

BAB II

TINJAUAN PUSTAKA

Bab ini berisi penjelasan teori-teori yang berkaitan dengan metode yang diajukan pada pengimplementasian perangkat lunak. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap sistem yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

2.1 Simulasi game

Sebuah game simulasi menggambarkan beragam super-kategori dari video game. Game simulasi umumnya dirancang untuk lebih dekat mensimulasikan kegiatan-kegiatan dunia nyata. Sebuah permainan simulasi mencoba untuk menyalin berbagai kegiatan dari kehidupan nyata dalam bentuk permainan untuk berbagai keperluan seperti pelatihan, analisis, atau prediksi. Biasanya tidak ada tujuan yang ditetapkan secara ketat dalam permainan, dengan pemain diizinkan untuk bebas mengontrol karakter [4].

Dari tiga jenis dasar permainan yaitu strategi, perencanaan, dan pembelajaran: permainan, simulasi, dan studi kasus, sejumlah gabungan dapat dipertimbangkan, termasuk permainan-permainan simulasi yang digunakan sebagai studi kasus. Perbandingan antara manfaat dari permainan simulasi dengan teknik pengajaran lainnya telah dilakukan oleh banyak peneliti dan sejumlah ulasan yang telah diterbitkan [5].

2.2 *Life simulation game*

Life simulation game adalah *subgenre* dari game simulasi dimana pemain tinggal atau mengontrol satu atau lebih jenis makhluk hidup virtual. Sebuah *life simulation game* dapat berbentuk individu-individu dan hubungan, atau bisa menjadi simulasi dari sebuah ekosistem.

Game simulasi kehidupan (*life simulation game*) adalah permainan tentang memelihara dan mengembangkan suatu populasi makhluk hidup yang dapat diatur, dimana pemain diberikan kekuatan untuk mengontrol kehidupan orang-orang *autonomous* atau makhluk hidup. Permainan kehidupan buatan terkait dengan penelitian ilmu komputer di bidang *artificial life*. Tapi karena permainan kehidupan buatan dimaksudkan untuk hiburan daripada penelitian, komersial permainan kehidupan menerapkan hanya sebagian dari apa penelitian kehidupan selidiki. Genre yang luas ini mencakup permainan dewa yang fokus pada pengelolaan suku penganut, serta binatang buatan yang fokus pada satu atau beberapa hewan. Ini juga termasuk permainan kehidupan buatan genetik, dimana pemain mengelola populasi makhluk hidup selama beberapa generasi [6][7].

2.3 *Survival game*

Survival game adalah subgenre dari video game *action* yang umumnya pemain memulai permainan dengan sumber daya minimal dalam keadaan bermusuhan, *open-world environment*, dan mengharuskan mereka untuk mengumpulkan sumber daya, alat-alat kerajinan, senjata, dan tempat tinggal, dan bertahan hidup selama mungkin. Banyak game survival didasarkan pada lingkungan persisten secara acak atau prosedural yang dihasilkan; akhir-akhir ini

permainan-permainan yang baru dibuat bisa dimainkan secara online dan memungkinkan banyak pemain untuk berinteraksi pada satu dunia yang tetap.

Survival game dianggap ekstensi dari tema video game pada umumnya dimana karakter pemain terdampar atau terpisah dari yang lain, dan harus bekerja sendiri untuk bertahan hidup dan menyelesaikan tujuan dari game. *Survival game* utamanya berfokus pada bagian bertahan hidup dari permainan ini, sementara mendorong eksplorasi dari *open-world*. *Survival game* adalah game *action*, meskipun beberapa elemen *gameplay* yang ada dalam genre *action-adventure*, seperti manajemen sumber daya dan barang-barang buatan, biasanya ditemukan di *survival game*, dan adalah elemen-elemen sentral dalam beberapa judul game. Pada *survival game*, umumnya pemain ditempatkan sendirian di dunia game dengan sumber daya yang sedikit. Hal ini tidak jarang membuat pemain menghabiskan sebagian atau keseluruhan dari permainan tanpa menemui karakter non-pemain yang baik; karena NPC biasanya bermusuhan dengan pemain, penekanan lebih ditekankan untuk menghindari daripada untuk konfrontasi. Dalam beberapa permainan, pertarungan tidak dapat dihindari dan menyediakan pemain dengan sumber daya yang berharga yaitu makanan, senjata, dan pakaian pelindung [7].

2.4 Finite State Machine

Finite State Machine adalah model matematis perhitungan yang digunakan untuk merancang kedua program komputer dan sirkuit logika sekuensial. Hal ini dipahami sebagai mesin abstrak yang dapat di salah satu dari sejumlah state terbatas. Mesin ini hanya dalam satu state pada suatu waktu; state itu pada waktu tertentu disebut

current state. Hal ini dapat berubah dari satu keadaan ke keadaan lain bila dimulai oleh sebuah external triggering event atau kondisi; ini disebut transisi. Sebuah FSM tertentu didefinisikan oleh daftar states yang, keadaan awal, dan kondisi memicu untuk setiap transisi.

Perilaku mesin-mesin state dapat diamati di banyak perangkat dalam masyarakat modern yang melakukan urutan yang telah ditentukan tindakan tergantung pada urutan peristiwa dengan yang mereka sajikan. contoh sederhana mesin penjual, yang mengeluarkan produk ketika kombinasi yang tepat dari koin yang dimasukkan, elevator, yang urutan berhenti ditentukan oleh lantai diminta oleh pengendara, lampu lalu lintas, yang mengubah urutan ketika mobil sedang menunggu, dan kombinasi kunci, yang membutuhkan masukan dari nomor kombinasi dalam urutan yang tepat. contoh yang lebih rumit adalah remote kontrol televisi dan video game, dan pengaturan waktu di jam tangan digital [8].

2.5 Unity

Unity adalah game engine pengembangan game yang terintegrasi kuat dengan satu set lengkap alat intuitif dan alur kerja yang cepat untuk membuat 3D interaktif dan konten 2D. Unity merupakan *easy multiplatform publishing*. Unity memiliki tool *Asset* yang menyediakan *Asset* untuk diunduh secara gratis maupun berbayar. Terdapat pula *Unity Community* yang menyediakan tutorial secara gratis untuk semua pengguna unity. Fitur scripting yang disediakan, mendukung tiga bahasa pemrograman, yaitu JavaScript, C#, dan Boo. *Flexible and EasyMoving, rotating, dan scaling objects* hanya perlu sebaris kode. Begitu juga dengan *Duplicating, removing, dan changing properties*. Visual Properties Variables yang

didefinisikan dengan scripts ditampilkan pada Editor. Bisa digeser, di *drag and drop*, bisa memilih warna dengan *color picker*. Berbasis .NET. Artinya penjalan program dilakukan dengan *Open Source .NET platform*, Mono. Serta mendukung pengembangan aplikasi Microsoft, SONY, Qualcomm, BlackBerry, Samsung, Nintendo, Oculus VR, dan Intel [3].

2.6 Android SDK

Sebuah kit pengembangan perangkat lunak yang memungkinkan developer untuk membuat aplikasi untuk platform android. Android SDK mencakupi contoh projek dengan source code, development tools, emulator, dan library yang dibutuhkan untuk membuat aplikasi android [9].

2.7 60 Seconds

60 seconds adalah permainan indie di mana pemain harus bertahan hidup dari pasca bom nuklir. Dalam permainan 60 seconds pemain memiliki 60 detik untuk mengais. Apapun barang (dan orang) yang berhasil didapatkan pemain dalam 60 detik tersebut akan menentukan nasib pemain. Setelah pemain dan siapa pun yang pemain pilih untuk dibawa dengan pemain, saat berada di bunker, anda akan melihat jurnal di kanan bawah layar anda. Buku itu pada dasarnya akan memberitahu anda segala sesuatu yang terjadi. Buku ini akan memberitahu anda ketika anda perlu menggunakan *supplies*, memberi makan *survivors* anda, dan apa yang terjadi di sekitar *shelter* anda.

(Halaman Ini Sengaja Dikosongkan)

BAB III

ANALISIS DAN PERANCANGAN

Pada bab ini akan dibahas mengenai analisis dan perancangan yang akan digunakan untuk menyelesaikan tugas akhir. Bab ini terdiri dari Analisis sistem, perancangan aplikasi realitas virtual, perancangan tampilan antar muka dan aturan.

3.1 Analisis Sistem

Aplikasi permainan simulasi bertahan hidup saat ini masih belum banyak ditemui, tapi semakin berkembang setiap harinya. Dan dengan pasar aplikasi yang menuntut untuk adanya aplikasi yang lebih baik, berbagai ide dapat diimplementasikan untuk membuat permainan simulasi bertahan hidup lebih menarik.

Aplikasi ini dibangun dengan tujuan mempelajari tingkah laku seseorang jika dihadapkan pada kondisi tertentu dan mengetahui seberapa lama seseorang dapat bertahan hidup di dalam permainan simulasi bertahan hidup ini

Dalam membangun aplikasi ini, penulis menggunakan *game engine* Unity versi 5.5.2f1 dengan bahasa pemrograman C# yang dibuat pada Visual Studio 2015. Untuk tampilan di dalam aplikasi, penulis menggunakan aset-aset gratisan yang dapat ditemukan di internet. Aplikasi ini dapat berjalan di *smartphone* Android.

3.2 Perancangan Aplikasi

3.2.1 Deskripsi Umum Perangkat Lunak

Tugas akhir yang akan dikembangkan adalah sebuah aplikasi permainan 2D bergenre simulasi dan *survival*. Tema aplikasi yang digunakan adalah bertahan hidup di pulau terpencil. Tempat yang dibuat dengan tema bertahan hidup di pulau terpencil adalah tempat pulau terpencilnya. Di dalam pulau terpencil tersebut terdapat 7 tempat yang bisa dipilih, yaitu *home*, hutan, padang rumput, sungai, pantai, gunung berapi, dan gua. Selain pulau terpencil, juga dibuat barang-barang yang bisa digunakan untuk bertahan hidup seperti makanan, minuman, panah, kapak, pisau, dan lain-lain.

Pada permainan simulasi bertahan hidup ini, Setiap harinya pemain melakukan kegiatan makan dan minum di pagi harinya dimana pemain bisa memilih untuk makan dan minum atau tidak di kegiatan tersebut. Setelah itu, pemain akan dihadapkan pada sebuah kejadian pada setiap waktu. Terkadang, pemain akan dihadapkan pada pilihan-pilihan pada sebuah event, dan pemain memilih salah satu dari pilihan yang ada. Setiap pilihan memiliki efek-efek hasil yang berbeda, dan pilihan yang dipilih akan menentukan kemana arah permainan si pemain berjalan

Pengguna utama dari aplikasi ini adalah semua orang yang memainkan permainan android. Terutama pemain yang ingin merasakan gimana bertahan hidup di pulau terpencil pada permainan ini.

3.2.2 Rancangan Penentuan Untuk Menang Dan Kalah

Setiap permainan memiliki penentuan untuk menang dan kalah di dalamnya. Hal tersebut juga berlaku untuk permainan simulasi bertahan hidup di pulau terpencil ini. Untuk memenangkan permainan simulasi bertahan hidup di pulau terpencil ini, pemain diharuskan untuk bertahan hidup selama 100 hari dan tidak ada satu status pun yang berada di tingkatan *dead*. Sedangkan pemain dianggap kalah jika salah satu diantara status *hungry*, *thirsty*, *life*, dan *sanity* mencapai *dead*, yaitu tingkatan paling bawah untuk setiap status.

3.2.3 Rancangan Faktor-Faktor Yang Terlibat Dalam Simulasi Bertahan Hidup

Pada permainan simulasi bertahan hidup ini ada beberapa faktor yang terlibat di dalamnya, yaitu status pemain, tempat yang bisa dikunjungi di pulau terpencil tersebut, waktu, dan kejadian-kejadian yang akan dialami si pemain selama bertahan hidup di pulau terpencil.

3.2.3.1 Status Pemain

Status pemain pada permainan simulasi bertahan hidup ini direpresentasikan dengan 4 status, yaitu *hungry*, *thirsty*, *life*, dan *sanity*.

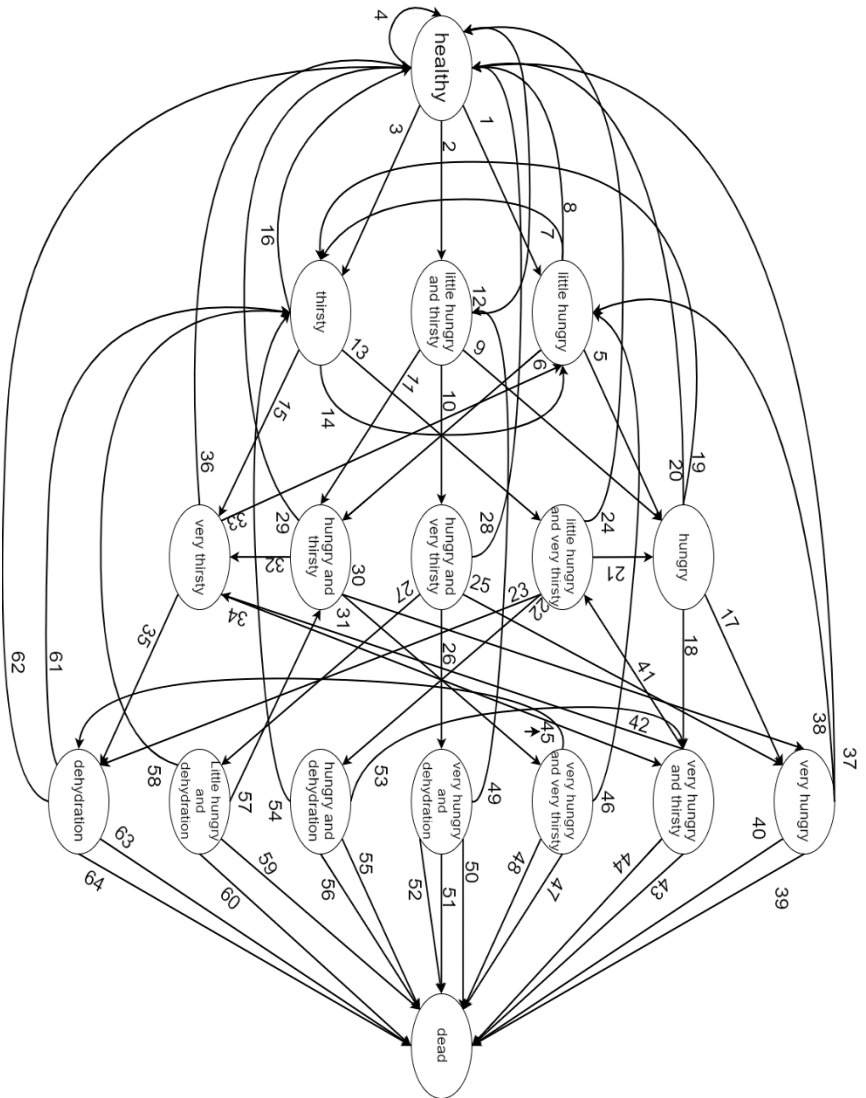
Hungry merepresentasikan tingkat kelaparan dari seorang pemain di permainan simulasi bertahan hidup ini. Sedangkan *thirsty* merepresentasikan tingkat kehausan dari seorang pemain di permainan simulasi bertahan hidup ini. Tingkatan paling tinggi untuk kedua status ini adalah *healthy*, sedangkan paling rendah untuk kedua status ini adalah *dead*. Di permainan ini, diasumsikan pemain adalah manusia normal, sehingga nilai maksimal status

hungry adalah 10 dan nilai maksimal status *thirsty* adalah 4. Daftar tingkatan status untuk *hungry* dan *thirsty* dapat dilihat pada Tabel 3.1.

Tabel 3.1Tingkatan status hungry dan thirsty

No	Nilai <i>Hungry</i>	Nilai <i>Thirsty</i>	Status
1	10	4	<i>Healthy</i>
2	10	3	<i>Thirsty</i>
3	10	2	<i>Very Thirsty</i>
4	10	1	<i>Dehydration</i>
5	10	0	<i>Dead</i>
6	7-9	4	<i>Little Hungry</i>
7	7-9	3	<i>Little Hungry and Thirsty</i>
8	7-9	2	<i>Little Hungry and Very Thirsty</i>
9	7-9	1	<i>Little Hungry and Dehydration</i>
10	7-9	0	<i>Dead</i>
11	4-6	4	<i>Hungry</i>
12	4-6	3	<i>Hungry and Thirsty</i>

13	4-6	2	<i>Hungry and Very Thirsty</i>
14	4-6	1	<i>Hungry and Dehydration</i>
15	4-6	0	<i>Dead</i>
16	1-3	4	<i>Very Hungry</i>
17	1-3	3	<i>Very Hungry and Thirsty</i>
18	1-3	2	<i>Very Hungry and Very Thirsty</i>
19	1-3	1	<i>Very Hungry and Dehydration</i>
20	1-3	0	<i>Dead</i>
21	0	4	<i>Dead</i>
22	0	3	<i>Dead</i>
23	0	2	<i>Dead</i>
24	0	1	<i>Dead</i>
25	0	0	<i>Dead</i>



Gambar 3.1 Finite state machine hungry dan thirsty

Pada saat pemain memulai permainan, status *hungry* dan *thirsty* selalu berawal di tingkat *healthy*. Lalu tingkatan tersebut akan berubah atau tetap di hari berikutnya tergantung dari input status yang dimasukkan di hari itu. Untuk *hungry* dan *thirsty*, input yang mengubah kedua status tersebut adalah pilihan dimana pemain memilih untuk makan atau tidak, dan minum atau tidak. Finite state machine dari *hungry* dan *thirsty* dapat dilihat di Gambar 3.1. Sedangkan penjelasan dari Finite state machine dari *hungry* dan *thirsty* di Gambar 3.1 dapat dilihat pada Tabel 3.2.

Tabel 3.2 Penjelasan finite state machine hungry dan thirsty

No	Dari State	Ke state	Input yang mengubah state
1	Healthy	Little hungry	Minum dan tidak makan
2	Healthy	Little hungry and thirsty	Tidak makan dan minum
3	Healthy	Thirsty	Makan dan tidak minum
4	Healthy	Healthy	Makan dan minum
5	Little hungry	Hungry	Minum dan tidak makan

6	Little hungry	Hungry and thirsty	Tidak makan dan minum
7	Little hungry	Thirsty	Makan dan tidak minum
8	Little hungry	Healthy	Makan dan minum
9	Little hungry and thirsty	Hungry	Minum dan tidak makan
10	Little hungry and thirsty	Hungry and very thirsty	Tidak makan dan minum
11	Little hungry and thirsty	Very thirsty	Makan dan tidak minum
12	Little hungry and thirsty	Healthy	Makan dan minum
13	Thirsty	Little hungry	Minum dan tidak makan
14	Thirsty	Little hungry and very thirsty	Tidak makan dan minum
15	Thirsty	Very thirsty	Makan dan tidak minum
16	Thirsty	Healthy	Makan dan minum

17	Hungry	Very hungry	Minum dan tidak makan
18	Hungry	Very hungry and thirsty	Tidak makan dan minum
19	Hungry	Thirsty	Makan dan tidak minum
20	Hungry	Healthy	Makan dan minum
21	Little hungry and very thirsty	Hungry	Minum dan tidak makan
22	Little hungry and very thirsty	Hungry and dehydration	Tidak makan dan minum
23	Little hungry and very thirsty	Dehydration	Makan dan tidak minum
24	Little hungry and very thirsty	Healthy	Makan dan minum
25	Hungry and very thirsty	Very hungry	Minum dan tidak makan
26	Hungry and very thirsty	Very hungry and dehydration	Tidak makan dan minum
27	Hungry and very thirsty	Dehydration	Makan dan tidak minum

28	Hungry and very thirsty	Healthy	Makan dan minum
29	Hungry and thirsty	Healthy	Makan dan minum
30	Hungry and thirsty	Very hungry	Minum dan tidak makan
31	Hungry and thirsty	Very hungry and very thirsty	Tidak makan dan minum
32	Hungry and thirsty	Very thirsty	Makan dan tidak minum
33	Very thirsty	Little hungry	Minum dan tidak makan
34	Very thirsty	Little hungry and dehydration	Tidak makan dan minum
35	Very thirsty	Dehydration	Makan dan tidak minum
36	Very thirsty	Healthy	Makan dan minum
37	Very hungry	Little hungry	Makan dan minum
38	Very hungry	Little hungry and thirsty	Makan dan tidak minum

39	Very hungry	Dead	Minum dan tidak makan
40	Very hungry	Dead	Tidak makan dan minum
41	Very hungry and thirsty	Little hungry and very thirsty	Makan dan tidak minum
42	Very hungry and thirsty	Little hungry	Makan dan minum
43	Very hungry and thirsty	Dead	Minum dan tidak makan
44	Very hungry and thirsty	Dead	Tidak makan dan minum
45	Very hungry and very thirsty	Little hungry and dehydration	Makan dan tidak minum
46	Very hungry and very thirsty	Little hungry	Makan dan minum
47	Very hungry and very thirsty	Dead	Minum dan tidak makan
48	Very hungry and very thirsty	Dead	Tidak makan dan minum
49	Very hungry and dehydration	Little hungry and thirsty	Makan dan minum

50	Very hungry and dehydration	Dead	Makan dan tidak minum
51	Very hungry and dehydration	Dead	Minum dan tidak makan
52	Very hungry and dehydration	Dead	Tidak makan dan minum
53	Hungry and dehydration	Very hungry and thirsty	Minum dan tidak makan
54	Hungry and dehydration	Thirsty	Makan dan minum
55	Hungry and dehydration	Dead	Makan dan tidak minum
56	Hungry and dehydration	Dead	Tidak makan dan minum
57	Little hungry and dehydration	Hungry and thirsty	Minum dan tidak makan
58	Little hungry and dehydration	Thirsty	Makan dan minum
59	Little hungry and dehydration	Dead	Makan dan tidak minum
60	Little hungry and dehydration	Dead	Tidak makan dan minum

61	Dehydration	Little hungry and thirsty	Minum dan tidak makan
62	Dehydration	Thirsty	Makan dan minum
63	Dehydration	Dead	Makan dan tidak minum
64	Dehydration	Dead	Tidak makan dan minum

Selain status *hungry* dan *thirsty*, permainan simulasi bertahan hidup ini juga memiliki status *sanity* dan *life*. Tingkatan paling tinggi untuk kedua status ini adalah *healthy*, sedangkan paling rendah untuk kedua status ini adalah *dead*. Daftar tingkatan status untuk *hungry* dan *thirsty* dapat dilihat pada Tabel 3.3. dan Tabel 3.4.

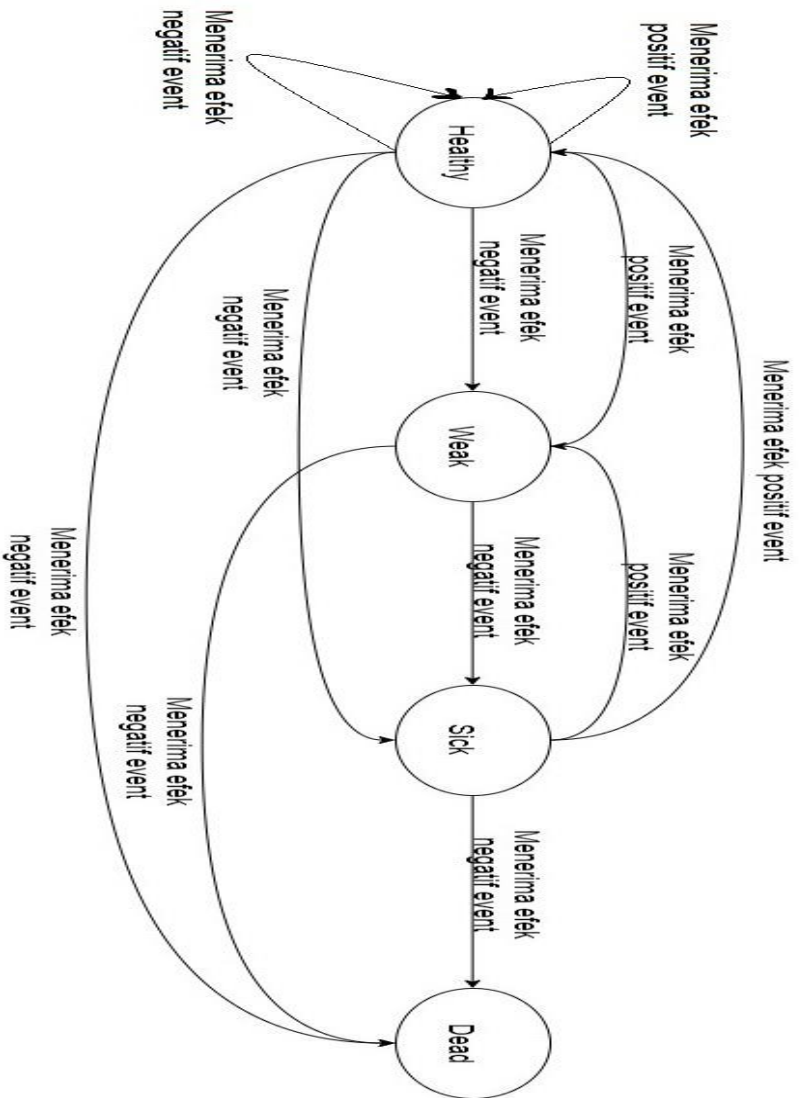
Tabel 3.3 Tingkatan status life

No	Nilai <i>Life</i>	Status
1	51-100	<i>Healthy</i>
2	26-50	<i>Weak</i>
3	1-25	<i>Sick</i>
4	0	<i>Dead</i>

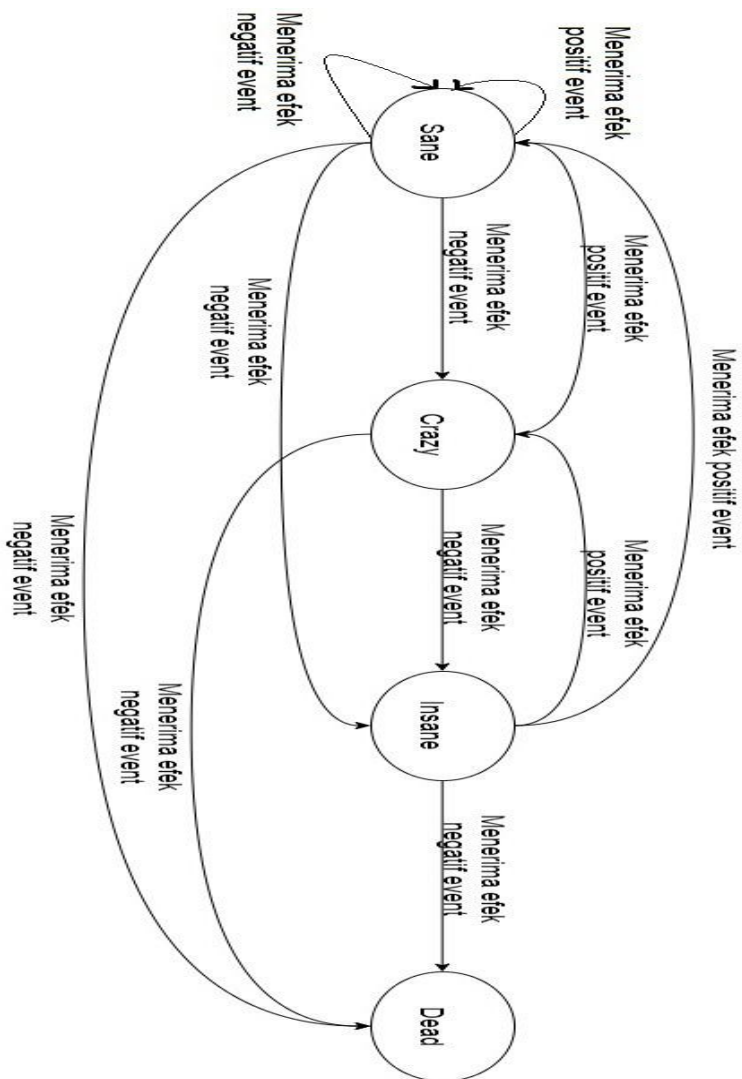
Tabel 3.4 Tingkatan status sanity

No	Nilai <i>Sanity</i>	Status
1	51-100	<i>Sane</i>
2	26-50	<i>Crazy</i>
3	1-25	<i>Insane</i>
4	0	<i>Dead</i>

Berbeda dengan parameter hungry dan thirsty, untuk parameter life dan sanity nilai parameternya tidak bertambah dan berkurang melalui makan dan minum, akan tetapi nilainya bertambah atau berkurang berdasarkan efek yang didapat dari pilihan dari suatu event yang diterima oleh pemain. Gambar dari *Finite State Machine* dari status *life* dan *sanity* dapat dilihat pada Gambar 3.2 dan Gambar 3.3.



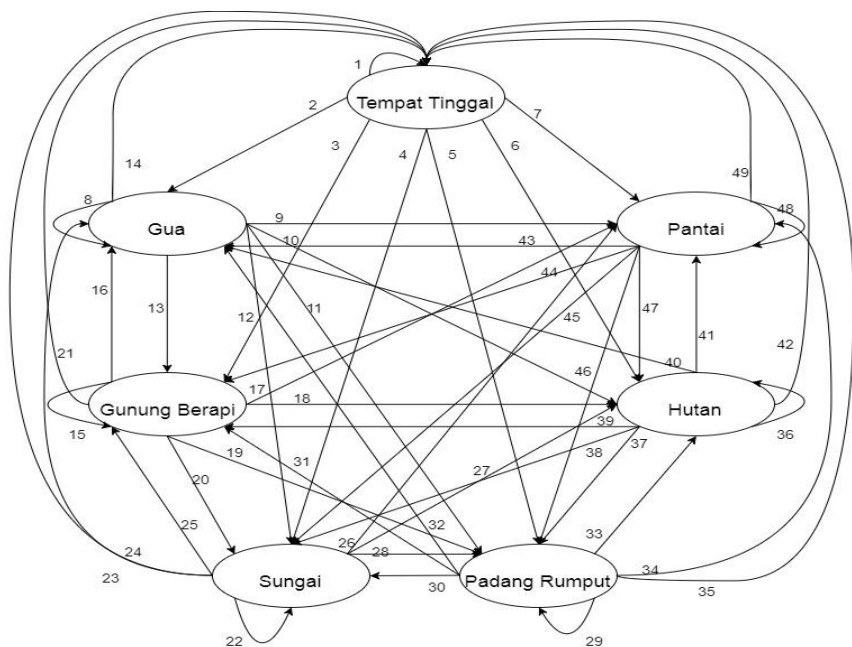
Gambar 3.2 Finite state machine life



Gambar 3.3 Finite state machine sanity

3.2.3.2 Tempat Yang Bisa Dikunjungi Di Pulau Terpencil

Pada permainan simulasi bertahan hidup di pulau terpencil ini, ada 7 tempat yang bisa dikunjungi pemain, yaitu *home*, hutan, padang rumput, sungai, pantai, gunung berapi, dan gua. Jika pemain memilih *home*, pemain langsung lanjut ke waktu berikutnya tanpa menerima event, sedangkan jika pemain memilih salah satu dari tempat selain *home*, pemain akan menerima event dari tempat yang dipilih. Finite state machine ini dapat dilihat pada Gambar 3.4, sedangkan penjelasannya dapat dilihat pada Tabel 3.5



Gambar 3.4 Finite state machine tempat

Tabel 3.5 Penjelasan finite state machine tempat

No	Dari state	Ke state	Input yang mengubah state
1	Tempat tinggal	Tempat tinggal	Pindah tempat/ganti hari
2	Tempat tinggal	Gua	Pindah tempat
3	Tempat tinggal	Gunung berapi	Pindah tempat
4	Tempat tinggal	Sungai	Pindah tempat
5	Tempat tinggal	Padang rumput	Pindah tempat
6	Tempat tinggal	Hutan	Pindah tempat
7	Tempat tinggal	Pantai	Pindah tempat
8	Gua	Gua	Pindah tempat
9	Gua	Pantai	Pindah tempat
10	Gua	Hutan	Pindah tempat
11	Gua	Padang rumput	Pindah tempat

12	Gua	Sungai	Pindah tempat
13	Gua	Gunung berapi	Pindah tempat
14	Gua	Tempat tinggal	Pindah tempat/ganti hari
15	Gunung berapi	Gunung berapi	Pindah tempat
16	Gunung berapi	Gua	Pindah tempat
17	Gunung berapi	Pantai	Pindah tempat
18	Gunung berapi	Hutan	Pindah tempat
19	Gunung berapi	Padang rumput	Pindah tempat
20	Gunung berapi	Sungai	Pindah tempat
21	Gunung berapi	Tempat tinggal	Pindah tempat/ganti hari
22	Sungai	Sungai	Pindah tempat
23	Sungai	Tempat tinggal	Pindah tempat/ganti hari

24	Sungai	Gua	Pindah tempat
25	Sungai	Gunung berapi	Pindah tempat
26	Sungai	Pantai	Pindah tempat
27	Sungai	Hutan	Pindah tempat
28	Sungai	Padang rumput	Pindah tempat
29	Padang rumput	Padang rumput	Pindah tempat
30	Padang rumput	Sungai	Pindah tempat
31	Padang rumput	Gunung berapi	Pindah tempat
32	Padang rumput	Gua	Pindah tempat
33	Padang rumput	Hutan	Pindah tempat
34	Padang rumput	Pantai	Pindah tempat
35	Padang rumput	Tempat tinggal	Pindah tempat/ganti hari

36	Hutan	Hutan	Pindah tempat
37	Hutan	Padang rumput	Pindah tempat
38	Hutan	Sungai	Pindah tempat
39	Hutan	Gunung berapi	Pindah tempat
40	Hutan	Gua	Pindah tempat
41	Hutan	Pantai	Pindah tempat
42	Hutan	Tempat tinggal	Pindah tempat/ganti hari
43	Pantai	Gua	Pindah tempat
44	Pantai	Gunung berapi	Pindah tempat
45	Pantai	Sungai	Pindah tempat
46	Pantai	Padang rumput	Pindah tempat
47	Pantai	Hutan	Pindah tempat
48	Pantai	Pantai	Pindah tempat
49	Pantai	Tempat tinggal	Pindah tempat/ganti hari

Pada awalnya pemain akan berada di tempat tinggal, lalu pemain memilih salah satu dari 7 tempat (termasuk tempat pemain berada sekarang). Setelah pemain sampai ditempat yang dipilih pemain menerima salah satu event berdasarkan dari tempat yang dipilih. Lalu saat berganti waktu (contoh dari pagi ke siang), posisi pemain sekarang berada di tempat yang dipilihnya di waktu sebelumnya. Lalu pemain memilih salah satu dari 7 tempat seperti sebelumnya, dan pemain menerima salah satu event dari tempat yang dipilihnya kembali, dan begitu seterusnya hingga hari berganti. Saat berganti hari pemain langsung kembali lagi ke tempat tinggal.

3.2.3.3 Waktu

Pada permainan simulasi bertahan hidup di pulau terpencil ini terdapat tiga waktu setiap harinya, yaitu pagi, siang, dan sore. Sedangkan malam hari langsung dilewati menuju ke hari selanjutnya. Waktu pagi direpresentasikan dengan bar berwarna biru, waktu siang direpresentasikan dengan bar berwarna kuning, dan waktu sore direpresentasikan dengan bar berwarna oranye.

3.2.3.4 Kejadian

Kejadian atau *event* adalah hal-hal yang terjadi pada pemain selama pemain bertahan hidup di pulau terpencil. Kejadian-kejadian yang akan dihadapi pemain tergantung dari tempat yang dipilih selain *home*. Meskipun begitu, kejadian yang akan dihadapi pemain bervariasi dan berbeda di setiap permainan

karena kejadian diberikan secara acak pada pemain. Kebanyakan kejadian di permainan simulasi bertahan hidup ini memiliki pilihan yang harus dipilih oleh pemain. Setiap pilihan memiliki efek masing-masing yang berbeda-beda, tapi secara umum efek pilihan dibagi menjadi 3, yaitu efek positif, efek netral, dan efek negatif. Contoh dari beberapa event di simulasi bertahan hidup ini beserta efek dari pilihannya dapat dilihat pada tabel 3.6.

Tabel 3.6 Contoh event dan efek pilihannya

No	Tempat	Event	Efek			
			Pilihan 1	Pilihan 2	Pilihan 3	Pilihan 4
1	Gua	Menjelajahi gua mengingatkanmu dengan keinginanmu di masa kecil yang ingin menjelajahi gua. Kamu menjelajahi gua dengan penuh semangat tanpa menemukan apapun	-	-	-	-

2	Gua	Saat kamu memeriksa gua, kamu menemukan sebuah kayu tertancap di dinding gua dan kamu menemukan hal itu aneh. Akankah kamu menarik kayu tersebut?	+1 tombak	Tidak ada	-	-
3	Gua	Kamu coba menjelajahi gua lebih dalam, tapi gua ini semakin gelap saat semakin dalam kamu jelajahi. Akankah kamu terus melanjutkan menjelajahi	-10 life	+1 Tali, (perlu obor)	Tidak ada	-

		gua ini atau tidak				
4	Gua	Saat kamu menjelajahi gua ini, kamu menemukan daerah yang kaya akan mineral-mineral tambang, tapi kamu perlu sesuatu untuk bisa menambang mineral-mineral tersebut	-100 life, -1 peledak (perlu peledak)	+1 bijih emas (perlu alat tambang)	Tidak ada	-
5	Gua	Saat kamu sedang menjelajahi gua, kamu menemukan seorang laki-laki sedang tergeletak. Kamu melihat laki-laki itu	+50 sanity, +1 alat tambang	+50 sanity, +2 makanan, +2 minuman, +1 alat tambang (perlu	-50 sanity, +2 makanan, +2 minuman	-10 sanity

		sedikit terluka. Akankah kamu menolong orang		first aid kit)		
6	Gunung berapi	Saat kamu mendaki gunung berapi, kamu menemukan pasir berwarna hitam yang tidak lain adalah bumbu mesiu. Kamu merasa kamu dapat membuat sesuatu dari bumbu mesiu itu...	+1 peledak (perlu buku)	+5 peluru (perlu buku)	Tidak ada	-
7	Gunung berapi	Kamu memutuskan untuk melihat	-	-	-	-

		pemandangan pulau ini dari tepi gunung berapi ini dan menikmatinya				
8	Gunung berapi	Kamu sedang melihat ke puncak gunung berapi. Tiba-tiba seorang wanita muda menyapamu. Melihat kamu sepertinya memiliki hasrat untuk mendaki gunung berapi tersebut, dia menawarkan alat-alat dari gunung buatannya padamu dan sebagai	+1 alat mendaki, -2 makanan (perlu minimal 2 makanan)	+1 alat mendaki, -2 minuman (perlu minimal 2 minuman)	Tidak ada	-

		gantinya dia menginginka n makanan atau minuman mu sebagai bayarannya				
9	Gunung berapi	Saat melihat lava yang mengalir dari gunung berapi, kamu teringat akan video-video dimana orang bermain dengan lava dari gunung berapi. Haruskah kamu mencoba hal tersebut atau tidak	+50 sanity, -25 life	Tidak ada	-	-
10	Gunung berapi	Ketika kamu berjalan di gunung berapi, kamu	+1 Obor (perlu kayu)	Tidak ada	-	-

		melihat lava yang mengalir dari gunung berapi. Kamu membutuhkan api untuk sesuatu. Apa yang akan kamu buat?				
11	Sungai	Saat kamu jalan-jalan ke sungai, kamu melihat ada banyak ikan di sungai dan cuacanya sedang bagus untuk menangkap ikan. Kamu memutuskan untuk...	Tidak ada	+1 makanan	+2 makanan (perlu alat pancing)	-
12	Sungai	Kamu merasa sangat haus dan menuju ke sungai untuk minum, saat	-20 life, +3 minuman	+2 minuman	Tidak ada	-

		kamu mau minum tiba-tiba ada seseorang menyerangmu. Apa yang akan kamu lakukan?				
13	Sungai	Kamu merasa sangat haus dan menuju ke sungai untuk minum, saat kamu mau minum tiba-tiba ada seseorang menyerangmu. Apa yang akan kamu lakukan?	-35 life, +2 minuman	-50 life	Tidak ada	-
14	Sungai	Kamu merasa bosan dan memutuskan untuk menghabiskan waktumu	-	-	-	-

		dengan melempar batu ke sungai				
15	Sung ai	Saat berjalan- jalan di tepi sungai, kamu menemukan seorang pria tergeletak di dekat sungai. Kamu mencoba menolongnya dan dia berkata bahwa dia kelaparan.	-1 makan an, +1 buku bertah an hidup (perlu paling tidak punya 1 makan an)	Tidak ada	-	-
16	Pada ng rump ut	Kamu berjalan- jalan di padang rump ut sambil merasakan angin segar di padang rump ut. Angin di	-	-	-	-

		padang rumput memang yang terbaik!				
17	Pada ng rumput	Kamu berjalan ke padang rumput dan menemukan sekelompok rusa disana. Apa yang akan kamu gunakan untuk memburu rusa-rusa tersebut	+2 makan an (perlu tomba k)	+4 makana n (perlu panah)	+4 makana n (perlu perang kap)	Tida k ada
18	Pada ng rumput	Saat berjalan di padang rumput, kamu menemukan tempat yang terlihat bisa digunakan untuk istirahat.	+25 life, +10 sanity	+2 makana n, -5 life	-	-

		Apakah kamu akan istirahat?				
19	Pada ng rumput	Melihat padang rumput yang luas ini, kamu entah mengapa teringat tentang cerita si penggembala domba. Mungkin kamu akan mencoba menggembala domba suatu hari nanti	-	-	-	-
20	Pada ng rumput	Saat kamu pulang melewati padang rumput, kamu melihat seorang wanita	Tidak ada	-40 life, +3 makanan	-15 life, +3 makanan, +1 kapak (perlu tombak)	+3 makanan, +1 senjata api, +5

		sedang bertarung dengan seekor macan...				peluru (perlengkapan)
21	Hutan	Saat kamu berkeliling di hutan, kamu menemukan kayu dan benang laba-laba yang sangat kuat. Sepertinya kamu dapat membuat sesuatu dari benda-benda tersebut. Apa yang akan kamu buat	+1 panah	+1 alat pancing	-	-
22	Hutan	Saat berjalan-jalan di hutan untuk mencari makanan, kamu	-20 life	Tidak ada	-	-

		menemukan jamur, tapi kamu tidak tahu apakah jamur ini beracun atau tidak? Haruskah kamu mencoba jamur tersebut atau tidak?				
23	Hutan	Saat berjalan-jalan di hutan untuk mencari makanan, kamu menemukan jamur, tapi kamu tidak tahu apakah jamur ini beracun atau tidak? Haruskah kamu mencoba jamur	+2 makanan	Tidak ada	-	-

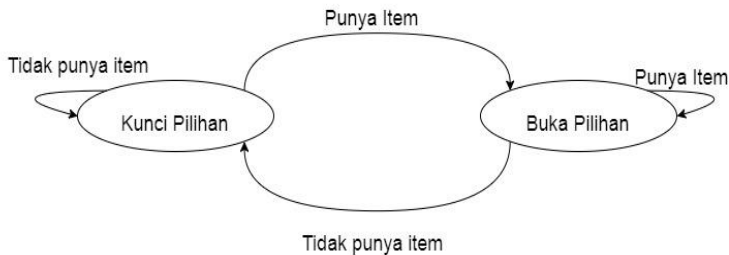
		tersebut atau tidak?				
24	Hutan	Saat sedang berjalan-jalan di hutan, kamu menemukan tanaman-tanaman yang bisa dibuat menjadi obat, tapi kamu tidak tahu cara membuat obat dari tanaman-tanaman tersebut	Tidak ada	+1 first aid kit (perlu buku bertahan hidup)	-	-
25	Hutan	Saat sedang berjalan-jalan di hutan, kamu menemukan tanaman-tanaman yang bisa	-20 life	+1 first aid kit (perlu buku bertahan hidup)	-	-

		dibuat menjadi obat, tapi kamu tidak tahu cara membuat obat dari tanaman-tanaman tersebut				
26	Pantai i	Waktu berjalan di pantai, kamu teringat akan masa kanak-kanak mu dimana kamu membuat sebuah istana pasir saat pergi ke pantai, dan kamu ada keinginan ingin membuat istana pasir lagi	+20 sanity	Tidak ada	-	-

27	Pantai	Kamu memutuskan untuk duduk santai di pantai sambil melihat ombak dan mendengarkan suara ombak	-	-	-	-
28	Pantai	Kamu menemukan seorang wanita tergeletak di pantai, dilihat dari keadaannya sepertinya dia sebelumnya tenggelam	Tidak ada	+20 sanity	-	-
29	Pantai	Saat kamu berenang di pantai untuk mencari ikan, tiba-tiba seekor ikan	-100 life	-30 life	-15 life, +8 makanan (perlu tombak)	-

		hiu menyerang kamu...				
30	Pantai	Sesampai di pantai, kamu memutuskan untuk bersantai di pantai. Kegiatan apa yang akan kamu lakukan	+20 sanity	+20 sanity	+20 sanity	-

Sedangkan gambar 3.5 menunjukkan finite state machine dari pilihan yang memerlukan barang di sebuah event.



Gambar 3.5 Finite state machine pilihan dengan barang

3.2.4 Perancangan Skenario Aplikasi

Alur skenario penggunaan dari aplikasi permainan simulasi bertahan hidup di pulau terpencil antara lain :

1. Pertama kali aplikasi dijalankan, pemain akan disajikan dengan tampilan menu utama. Pada tampilan itu terdapat 3 tombol, yaitu *new game*, *how to play*, dan *quit*. Pengguna memilih *new game* untuk memulai permainan.
2. Masuk ke dunia permainan, pemain masuk ke bagian awal yang dimana pemain diperlihatkan pulau terpencilnya dan status si pemain. Selain itu, pemain diberikan 2 tombol yang dapat dipilih, yaitu tombol *inventory*, *pause*, dan *makan minum*. Jika pemain memilih tombol *inventory*, pemain masuk ke menu *inventory*. Jika pemain memilih tombol *event*, pemain masuk ke bagian makan dan minum. Jika pemain memilih tombol *pause*, pemain masuk ke menu *pause*. Pada bagian ini, pemain memilih tombol *event* untuk masuk ke bagian makan dan minum
3. Di bagian makan dan minum, pemain dapat memilih untuk makan dan/atau minum pada hari itu. Setelah selesai memilih, pemain menekan tombol panah ke kanan untuk masuk ke bagian *event*
4. Di bagian *event*, pemain dapat memilih 7 tempat yang dapat dikunjungi di pulau terpencil. Jika pemain memilih tempat tinggal/*home*, pemain lanjut ke waktu berikutnya atau ke hari berikutnya jika memilihnya di sore hari. Jika pemain memilih tempat selain tempat tinggal/*home*, pemain akan diberikan kejadian sesuai dengan tempat yang dipilih pemain. Jika kejadian tersebut memiliki pilihan, maka pemain harus memilih salah satu dari pilihan yang ada di kejadian tersebut. Setelah memilih, pemain menerima efek dari pilihan yang dipilih oleh pemain dan lanjut ke waktu berikutnya hingga sore

hari. Setelah sore hari selesai, pemain masuk ke hari berikutnya dan kembali lagi ke bagian awal.

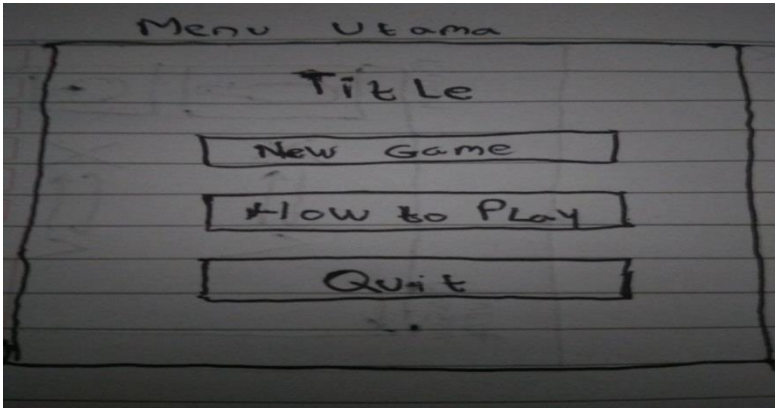
5. Pemain mengulangi skenario dari nomor 2 hingga nomor 4 sampai pemain bertahan hidup selama 100 hari dan memenangkan permainan atau pemain mendapatkan *game over*.

3.3 Perancangan Tampilan Antarmuka

Subbab ini membahas bagaimana rancangan antarmuka pengguna yang akan digunakan untuk tugas akhir. Dalam aplikasi ini terdapat beberapa tampilan, yaitu tampilan menu utama, bagian awal, bagian makan dan minum, dan bagian event, menu inventory, dan menu pause

3.3.1 Tampilan Menu Utama

Tampilan Menu Utama pada Gambar 3.6 merupakan tampilan yang pertama kali muncul ketika aplikasi dijalankan. Pada tampilan menu utama terdapat 3 button, yaitu *new game*, *how to play*, dan *quit*.



Gambar 3.6 Rancangan tampilan menu utama

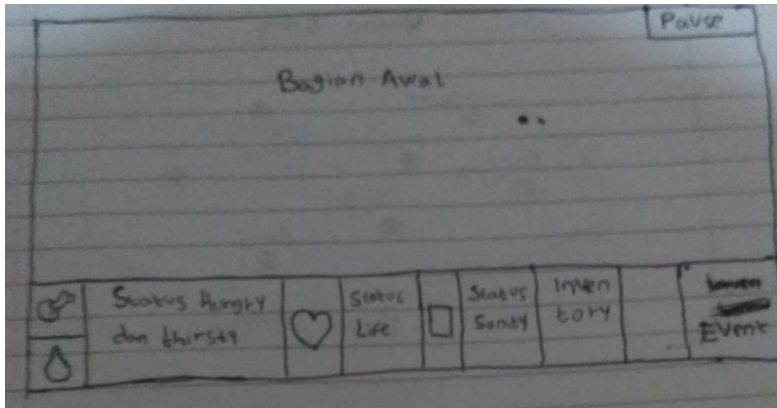
Seperti yang terlihat pada Gambar 3.6

1. Button *new game* digunakan untuk pemain memulai sesi permainan yang baru di permainan simulasi bertahan hidup di pulau terpencil
2. Button *how to play* digunakan untuk menuju menu tentang cara bermain permainan simulasi bertahan hidup di pulau terpencil.
3. Button *quit* digunakan untuk pemain keluar dari aplikasi permainan simulasi bertahan hidup di pulau terpencil.

3.3.2 Tampilan Bagian Awal

Tampilan bagian awal pada Gambar 3.7 merupakan tampilan yang pertama kali muncul saat pemain memulai sesi permainan yang baru di permainan simulasi bertahan hidup di pulau terpencil ini. Di dalam tampilan ini terdapat informasi status

hungry, *thirsty*, *life*, dan *sanity* pemain, serta button *pause*, *inventory*, dan *event*.



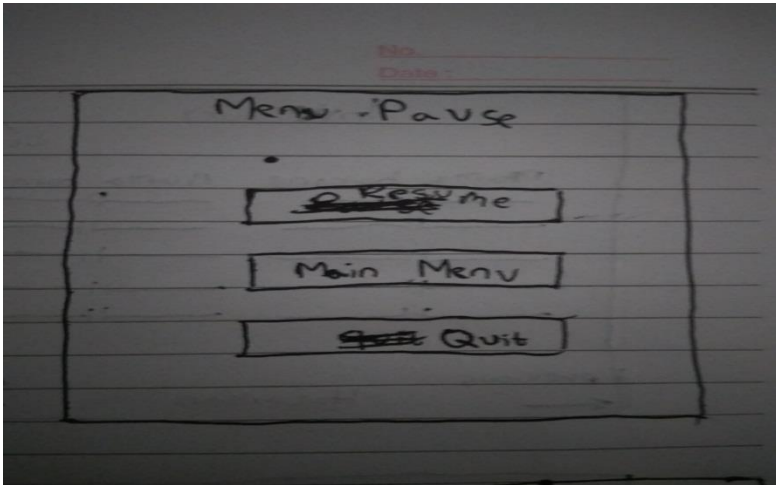
Gambar 3.7 Rancangan antarmuka bagian awal

Seperti yang terlihat pada Gambar 3.7

1. Button *pause* digunakan untuk pemain masuk ke menu *pause*.
2. Button *inventory* digunakan untuk pemain masuk ke menu *inventory*.
3. Button *event* digunakan untuk pemain masuk ke bagian makan dan minum.

3.3.3 Tampilan Menu Pause

Tampilan menu pause pada Gambar 3.8 merupakan tampilan yang digunakan untuk pemain kembali ke menu utama atau keluar dari permainan saat sesi permainan sedang berlangsung. Di dalam tampilan ini terdapat 3 button, yaitu *resume*, *main menu*, dan *quit*.



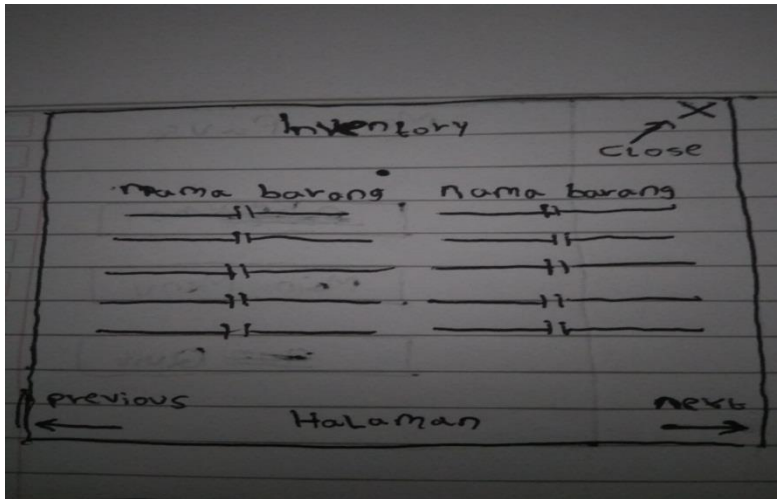
Gambar 3.8 Rancangan antarmuka menu pause

Seperti yang terlihat pada Gambar 3.8

1. Button *resume* digunakan untuk pemain melanjutkan sesi permainan yang dilakukannya.
2. Button *main menu* digunakan untuk pemain kembali ke menu utama saat sesi permainan berlangsung.
3. Button *quit* digunakan untuk pemain keluar dari aplikasi permainan simulasi bertahan hidup di pulau terpencil.

3.3.4 Tampilan Menu Inventory

Tampilan menu *inventory* pada Gambar 3.9 merupakan tampilan yang digunakan untuk melihat barang-barang yang dimiliki si pemain saat pemain berada di dalam sesi permainan. Di dalam tampilan ini terdapat 3 button, yaitu *previous*, *next*, dan *close*.



Gambar 3.9 Rancangan antarmuka menu inventory

Seperti yang terlihat pada Gambar 3.9

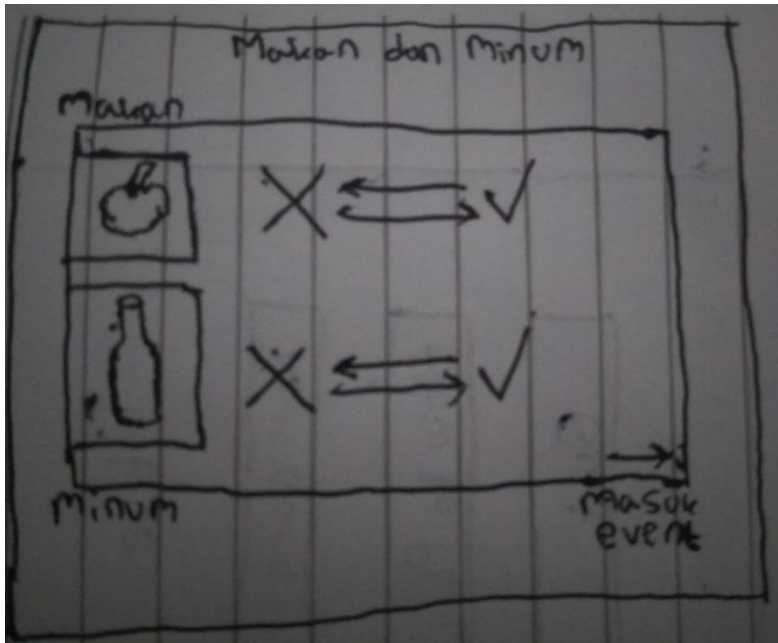
1. Button *previous* digunakan untuk kembali ke halaman *inventory* sebelumnya. Button ini cuma akan muncul jika pemain tidak berada di halaman awal dan jumlah halaman *inventory* lebih dari 1. Button ini direpresentasikan dengan gambar panah ke kiri
2. Button *next* digunakan untuk lanjut ke halaman *inventory* selanjutnya. Button ini cuma akan muncul jika pemain tidak berada di halaman akhir dan jumlah halaman *inventory* lebih dari 1. Button ini direpresentasikan dengan gambar panah ke kanan
3. Button *close* digunakan untuk pemain keluar dari menu *inventory* dan kembali ke tampilan bagian awal. Button ini direpresentasikan dengan gambar silang

3.3.5 Tampilan Bagian Makan Dan Minum

Tampilan bagian makan dan minum pada Gambar 3.10 merupakan tampilan dimana pemain memilih untuk makan dan minum sebelum memulai kejadian atau *event* pada hari tersebut. Pada tampilan ini terdapat 3 button, yaitu makan, minum, dan masuk *event*.

Seperti yang terlihat pada Gambar 3.10

1. Button makan digunakan pemain untuk memilih untuk makan atau tidak. Button ini direpresentasikan dengan gambar apel. Sedangkan pemain memilih untuk makan direpresentasikan dengan tanda benar dan pemain memilih untuk tidak makan direpresentasikan dengan tanda silang.
2. Button minum digunakan pemain untuk memilih untuk minum atau tidak. Button ini direpresentasikan dengan gambar botol minuman. Sedangkan pemain memilih untuk minum direpresentasikan dengan tanda benar dan pemain memilih untuk tidak minum direpresentasikan dengan tanda silang.
3. Button masuk *event* digunakan pemain untuk masuk ke bagian *event*. Button ini direpresentasikan dengan gambar panah ke kanan



Gambar 3.10 Rancangan antarmuka bagian makan dan minum

3.3.6 Tampilan Bagian Event

Tampilan bagian *event* pada Gambar 3.11 merupakan tampilan dimana pemain menerima kejadian atau event. Pada tampilan ini di kanan atas terdapat 3 bar waktu yang menunjukkan waktu di hari tersebut. Selain itu di tampilan ini terdapat 7 button, yaitu tempat tinggal atau *home*, hutan, padang rumput, sungai, pantai, gunung berapi, dan gua.

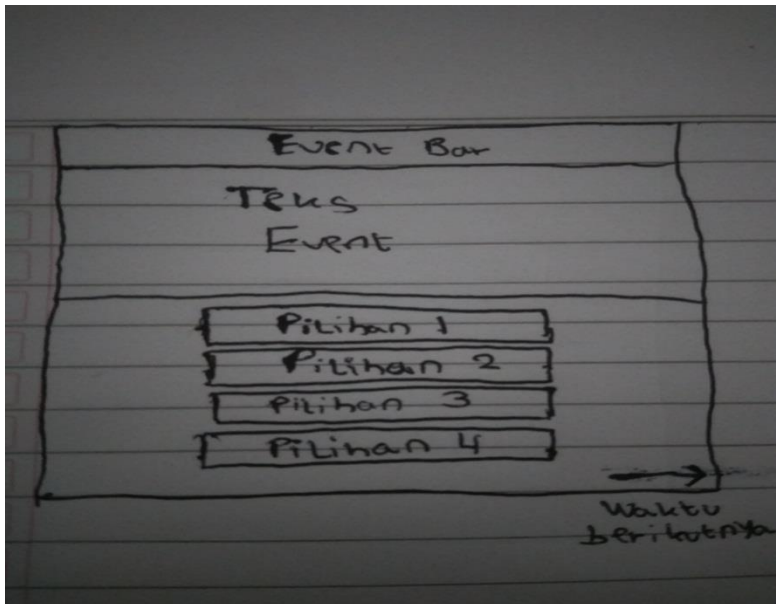


Gambar 3.11 Rancangan antarmuka bagian event

Seperti yang terlihat pada Gambar 3.11

1. Button tempat tinggal atau *home* digunakan pemain untuk lanjut ke waktu berikutnya tanpa menerima event.
2. Button hutan digunakan untuk menerima event dari tempat hutan.
3. Button padang rumput digunakan untuk menerima event dari tempat padang rumput.
4. Button sungai digunakan untuk menerima event dari tempat sungai.
5. Button pantai digunakan untuk menerima event dari tempat pantai.
6. Button gunung berapi digunakan untuk menerima event dari tempat gunung berapi.
7. Button gua digunakan untuk menerima event dari tempat gua.

Jika pemain memilih tempat selain home, maka akan muncul *event bar* seperti pada Gambar 3.12. Di dalam *event bar*, terdapat teks yang berisi event yang diterima pemain beserta button pilihan-pilihannya (jika ada), dan button ke waktu berikutnya.



Gambar 3.12 Rancangan event bar

Seperti yang terlihat pada Gambar 3.12

1. Button Pilihan 1 untuk pemain memilih pilihan ke-1 dari sebuah event. Button pilihan 1 cuma muncul jika *event* mempunyai pilihan 1
2. Button Pilihan 2 untuk pemain memilih pilihan ke-2 dari sebuah event. Button pilihan 2 cuma muncul jika *event* mempunyai pilihan 1

3. Button Pilihan 3 untuk pemain memilih pilihan ke-3 dari sebuah event. Button pilihan 3 cuma muncul jika *event* mempunyai pilihan 3
4. Button Pilihan 4 untuk pemain memilih pilihan ke-4 dari sebuah event. Button pilihan 4 cuma muncul jika *event* mempunyai pilihan 4
5. Button waktu berikutnya digunakan pemain untuk lanjut ke waktu berikutnya. Button ini cuma muncul setelah pemain memilih pilihan dari sebuah event atau jika *event* tersebut tidak memiliki pilihan.
6. Teks *event* berisi penjelasan event yang diterima oleh pemain.

BAB IV

IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi dari perancangan aplikasi. Di dalamnya mencakup proses penerapan dan pengimplementasian dalam bentuk skenario, dan antarmuka yang mengacu pada rancangan yang telah dibahas sebelumnya.

4.1 Lingkungan Implementasi

Lingkungan implementasi dari tugas akhir ini adalah perangkat *smartphone* dengan sistem operasi Android.

4.2 Implementasi Antarmuka

Berikut implementasi antarmuka main menu pada Gambar 4.1, antarmuka bagian awal pada Gambar 4.2, antarmuka menu pause pada Gambar 4.3, antarmuka menu inventory pada Gambar 4.4, antarmuka bagian makan dan minum pada Gambar 4.5, antarmuka bagian event pada Gambar 4.6, dan *event bar* pada Gambar 4.7.



Gambar 4.1 Antarmuka menu utama



Gambar 4.2 Antarmuka bagian awal



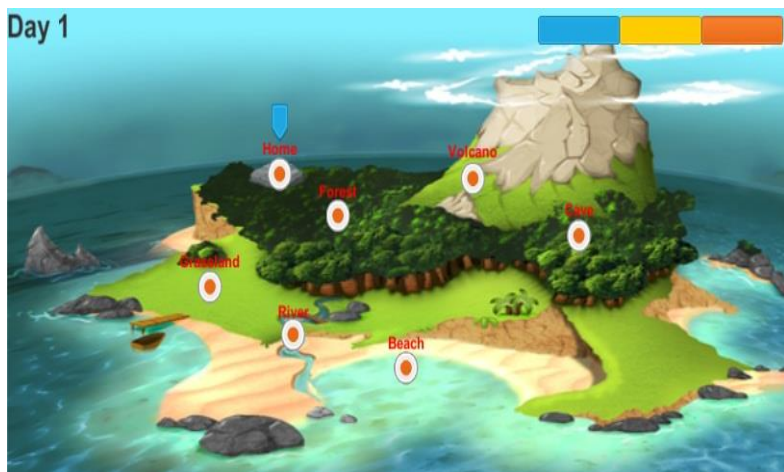
Gambar 4.3 Antarmuka menu pause



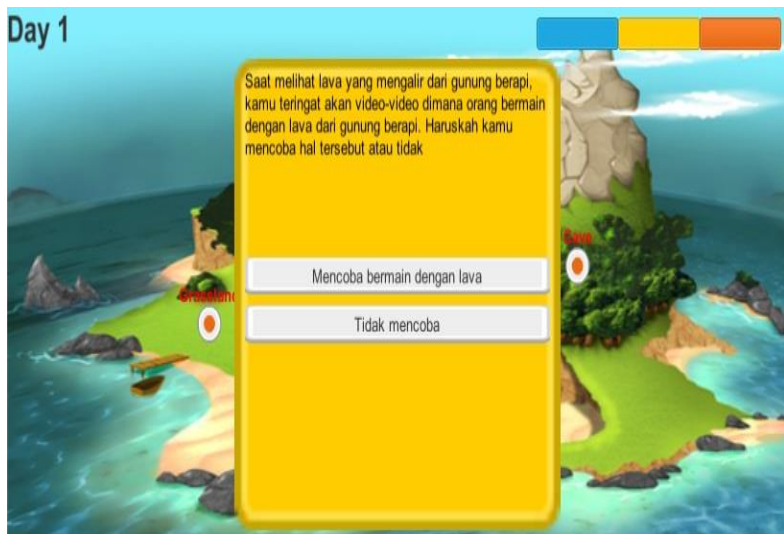
Gambar 4.4 Antarmuka menu inventory



Gambar 4.5 Antarmuka bagian makan dan minum



Gambar 4.6 Antarmuka bagian event



Gambar 4.7 Antarmuka event bar

4.3 Implementasi Faktor-Faktor Bertahan Hidup dan Finite State Machine

Masing-masing faktor-faktor bertahan hidup dan *finite state machine* yang dibuat diimplementasikan menjadi acuan unsur simulasi pada aplikasi permainan simulasi bertahan hidup di pulau terpencil ini.

4.3.1 Penentuan Untuk Menang dan Kalah

Seperti yang ditulis sebelumnya, Setiap permainan memiliki penentuan untuk menang dan kalah didalamnya. Untuk memenangkan permainan simulasi bertahan hidup di pulau terpencil ini, pemain diharuskan untuk bertahan hidup selama 100 hari. Sedangkan pemain dianggap kalah jika salah satu diantara status *hungry*, *thirsty*, *life*, dan *sanity* mencapai *dead*, yaitu tingkatan paling bawah untuk setiap status. Gambar dari saat pemain memenangkan permainan simulasi ini dapat dilihat pada Gambar 4.8. Sedangkan gambar dari saat pemain kalah dapat dilihat pada Gambar 4.9. Kode program dari pengecekan untuk melihat apakah pemain sudah memenuhi syarat menang atau kalah atau tetap melanjutkan permainan dapat dilihat pada Kode Sumber 4.1.



Gambar 4.8 Scene pemain menang



Gambar 4.9 Scene pemain kalah

```
switch (Status.characterlife)
{
    case true:
        switch (Status.day)
        {
            case 100:
                SceneManager.LoadScene("Scene Menang");
                break;
            default:
                SceneManager.LoadScene("Scene Utama");
                break;
        }
        break;
    case false:
        SceneManager.LoadScene("Scene Game Over");
        break;
}
```

Kode Sumber 4.1 Penentuan menang dan kalah

4.3.2 Status

Di dalam permainan simulasi bertahan hidup di pulau terpencil ini, pemain memiliki status *hungry*, *thirsty*, *life*, dan *sanity*. Saat pemain memulai sesi permainan, pemain secara *default* memiliki status *hungry* dan *thirsty* berada di tingkat *healthy*, status *life* di tingkat *healthy*, dan status *sanity* di tingkat *sane* seperti yang terlihat pada gambar 4.10. Kode program untuk mengatur status *hungry*, *thirsty*, *life*, dan *sanity* dapat dilihat pada Kode Sumber 4.2 dan Kode Sumber 4.3. Kode program untuk menampilkan status *hungry* dan *thirsty* dapat dilihat pada Kode Sumber 4.4. Kode program untuk menampilkan status *life* dapat dilihat pada Kode Sumber 4.5. Kode program untuk menampilkan status *sanity* dapat dilihat pada Kode Sumber 4.6.



Gambar 4.10 Status saat memulai permainan

```

public class Status
{
    public static int hungry = 4;
    public static bool stathungry = true;
    public static int thirsty = 4;
    public static bool statthirsty = true;
    public static int life = 100;
    public static bool statlife = true;
    public static int sanity = 100;
    public static bool statsanity = true;

    public static void SetStatusUlang()
    {
        hungry = 4;
        stathungry = true;
        thirsty = 4;
        statthirsty = true;
        life = 100;
        statlife = true;
        sanity = 100;
        statsanity = true;
    }
}

```

Kode Sumber 4.2 Mengatur status

```

public class MainMenu : MonoBehaviour
{
    void Start()
    {
        Status.SetStatusUlang();
    }
}

```

Kode Sumber 4.3 Mengatur status saat memulai permainan

```

using UnityEngine.UI;

public class Utama : MonoBehaviour
{
    void Start()
    {
        int hungry = Status.hungry;
        int thirsty = Status.thirsty;
        GameObject status_HungryThirsty =
this.gameObject.transform.GetChild(1).GetChild(2).gameObject;

        switch (hungry)
        {
            case 4:
                switch (thirsty)
                {
                    case 4:
                        status_HungryThirsty.GetComponent<Text>().text = "Healthy";
                        break;
                    case 3:
                        status_HungryThirsty.GetComponent<Text>().text = "Thirsty";
                        break;
                    case 2:
                        status_HungryThirsty.GetComponent<Text>().text = "Very thirsty";
                        break;
                    case 1:
                        status_HungryThirsty.GetComponent<Text>().text = "Dehydration";
                        break;
                    case 0:
                        status_HungryThirsty.GetComponent<Text>().text = "Dead";
                        break;
                }
                break;
            case 3:
                switch (thirsty)
                {
                    case 4:
                        status_HungryThirsty.GetComponent<Text>().text = "Little hungry";
                        break;
                    case 3:
                        status_HungryThirsty.GetComponent<Text>().text = "Little hungry
and thirsty";
                        break;

```

Kode Sumber 4.4 Status hungry dan thirsty

```

using UnityEngine.UI;

public class Utama : MonoBehaviour
{
    void Start()
    {
        int life = Status.life;
        GameObject status_Life =
this.gameObject.transform.GetChild(1).GetChild(4).gameObject;

        switch(life){
            case 100:
            case 95:
            case 90:
            case 85:
            case 80:
            case 75:
            case 70:
            case 65:
            case 60:
            case 55:
                status_Life.GetComponent<Text>().text = "Healthy";
                break;
            case 50:
            case 45:
            case 40:
            case 35:
            case 30:
                status_Life.GetComponent<Text>().text = "Weak";
                break;
            case 25:
            case 20:
            case 15:
            case 10:
            case 5:
                status_Life.GetComponent<Text>().text = "Sick";
                break;
            case 0:
                status_Life.GetComponent<Text>().text = "Dead";
                break;
        }
    }
}

```

Kode Sumber 4.5 Status life

```

using UnityEngine.UI;

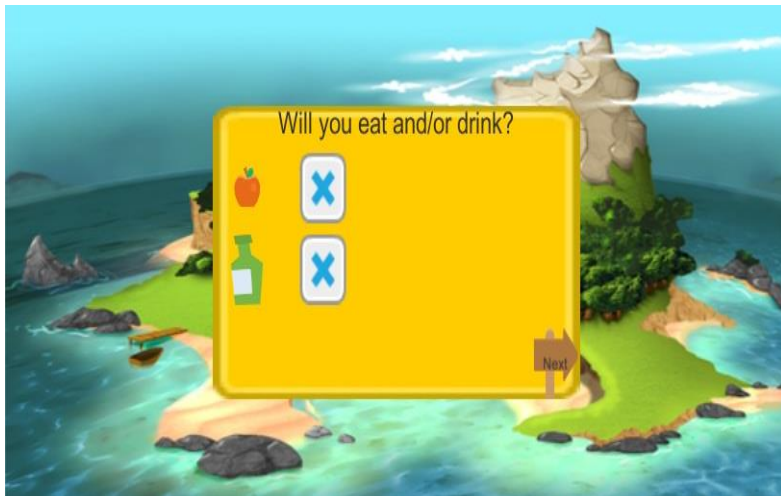
public class Utama : MonoBehaviour
{
    void Start()
    {
        int sanity = Status.sanity;
        GameObject status_Sanity =
this.gameObject.transform.GetChild(1).GetChild(6).gameObject;

        switch(life){
            case 100:
            case 95:
            case 90:
            case 85:
            case 80:
            case 75:
            case 70:
            case 65:
            case 60:
            case 55:
                status_Life.GetComponent<Text>().text = "Sane";
                break;
            case 50:
            case 45:
            case 40:
            case 35:
            case 30:
                status_Life.GetComponent<Text>().text = "Crazy";
                break;
            case 25:
            case 20:
            case 15:
            case 10:
            case 5:
                status_Life.GetComponent<Text>().text = "Insane";
                break;
            case 0:
                status_Life.GetComponent<Text>().text = "Dead";
                break;
        }
    }
}

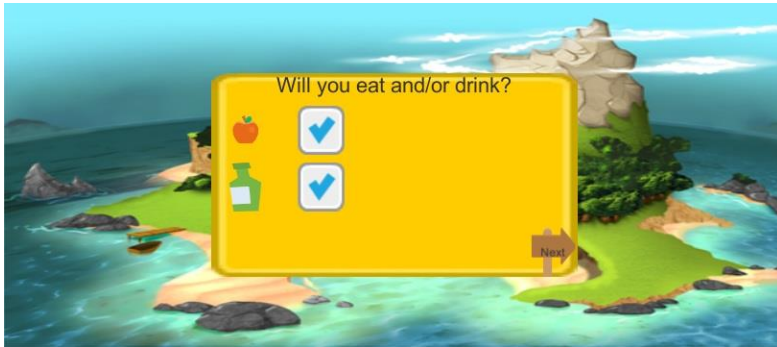
```

Kode Sumber 4.6 Status sanity

Setiap harinya, pemain diharuskan untuk melakukan kegiatan makan dan minum. Saat melakukan kegiatan makan dan minum, Pada awalnya pemain secara default dipilhkan untuk tidak makan dan tidak minum. Pemain dapat menggantinya menjadi memilih makan dan memilih minum, tapi pemain paling tidak harus memiliki paling sedikit 1 makanan untuk bisa ganti menjadi memilih makan. Begitu juga untuk minum, pemain paling tidak harus memiliki paling sedikit 1 minuman untuk bisa ganti menjadi memilih minum. Kondisi awal saat makan dan minum dapat dilihat pada Gambar 4.11, sedangkan kondisi dimana pemain memilih makan dan minum dapat dilihat pada Gambar 4.12. Kode program untuk memilih makan dapat dilihat pada Kode Sumber 4.7 dan kode program untuk memilih minum dapat dilihat pada Kode Sumber 4.8.



Gambar 4.11 Kondisi awal makan dan minum



Gambar 4.12 Kondisi pemain memilih makan dan minum

```
public class MakanDanMinum : MonoBehaviour
{
    private GameObject CrossMakan;
    private GameObject CheckMakan;
    void Start()
    {
        CheckMakan = this.gameObject.transform.GetChild(1).GetChild(2);
        CheckMakan.SetActive(false);
    }

    public void ButtonMakan()
    {
        switch(CheckMakan.activeInHierarchy)
        {
            case false:
                if (Barang.makanan >= 1)
                {
                    CheckMakan.SetActive(true);
                }
                else
                {
                    CheckMakan.SetActive(false);
                }
                break;
            case true:
                CheckMakan.SetActive(false);
        }
    }
}
```

Kode Sumber 4.7 Memilih makan

```

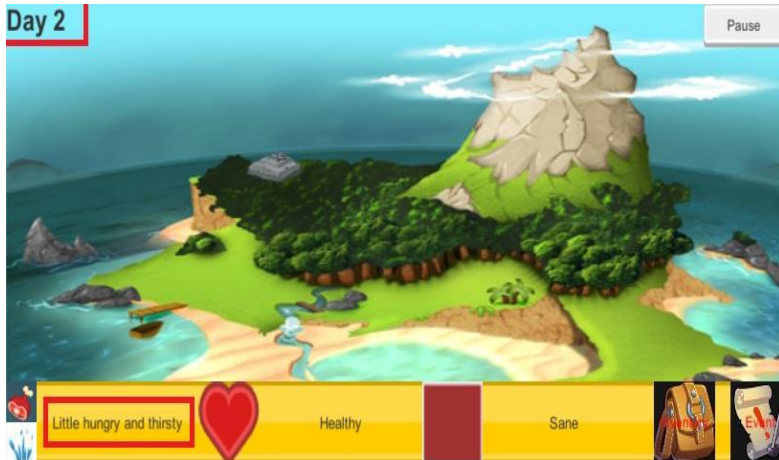
public class MakanDanMinum : MonoBehaviour
{
    private GameObject CrossMinum;
    private GameObject CheckMinum;
    void Start()
    {
        CheckMinum =
this.gameObject.transform.GetChild(1).GetChild(4).gameObject;
        CheckMinum.SetActive(false);
    }

    public void ButtonMinum()
    {
        switch (CheckMinum.activeInHierarchy)
        {
            case false:
                if (Barang.minuman >= 1)
                {
                    CheckMinum.SetActive(true);
                }
                else
                {
                    CheckMinum.SetActive(false);
                }
                break;
            case true:
                CheckMinum.SetActive(false);
                break;
        }
    }
}

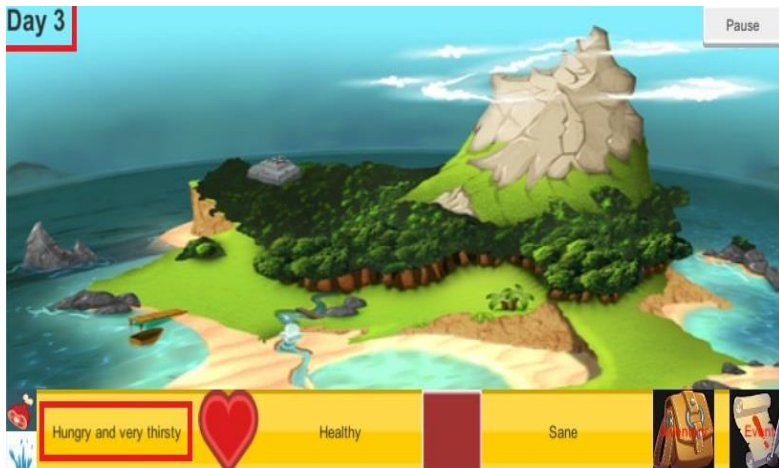
```

Kode Sumber 4.8 Memilih minum

Setelah pemain memilih pilihannya di makan dan minum dan masuk ke bagian *event*, aplikasi mulai mengecek input makan dan minum pemain, dan mengganti status hungry dan thirsty sesuai input yang dipilih pemain. Contoh status pemain setelah memilih untuk tidak makan dan tidak minum dapat dilihat pada Gambar 4.13. Kode program pengecekan input makan dapat dilihat pada Kode Sumber 4.9 dan kode program pengecekan input minum dapat dilihat pada Kode Sumber 4.10.



Gambar 4.13 Status hungry dan thirsty setelah memilih tidak makan dan tidak minum



Gambar 4.14 Status hungry dan thirsty setelah memilih tidak makan dan tidak minum (2)

```

public void OpenEvent()
{
    switch(CheckMakan.activeInHierarchy)
    {
        case true:
            Status.hungry = Status.hungry + 2;
            if (Status.hungry > 4)
            {
                Status.hungry = 4;
            }
            Barang.makanan--;
            if (Barang.makanan == 0)
            {
                Barang.makananstatus = false;

                Barang.ListBarang.RemoveAt(Barang.ListBarang.IndexOf("Makanan"));
            }
            break;
        case false:
            Status.hungry--;
            if (Status.hungry == 0)
            {
                Status.characterlife = false;
                Status.stathungry = false;
            }
            break;
    }
}

```

Kode Sumber 4.9 Pengecekan makan

```

public void OpenEvent()
{
    switch (CheckMinum.activeInHierarchy)
    {
        case true:
            Status.thirsty = Status.thirsty + 2;
            if (Status.thirsty > 4)
            {
                Status.thirsty = 4;
            }
            Barang.minuman--;
            if (Barang.minuman == 0)
            {
                Barang.minumanstatus = false;

                Barang.ListBarang.RemoveAt(Barang.ListBarang.IndexOf("Minuman"));
            }
            break;
        case false:
            Status.thirsty--;
            if (Status.thirsty == 0)
            {
                Status.characterlife = false;
                Status.statthirsty = false;
            }
            break;
    }
}

```

Kode Sumber 4.10 Pengecekan minum

Status *hungry* dan *thirsty* menerima input dari kegiatan makan dan minum, tapi tidak untuk status *life* dan *sanity*. status *life* dan *sanity* menerima input untuk mengubah nilainya dari pilihan yang dipilih oleh pemain saat menerima event. Efek pilihan dari pilihan yang dipilih pemain dapat yang mengurangi atau menambah nilai status *life*, dan mengurangi atau menambah nilai status *sanity*. Kode program salah satu contoh perubahan nilai status *life* dan *sanity* dapat dilihat pada Kode Sumber 4.11

```

public class EventGunungBerapi2
{
    public EventGunungBerapi2()
    {
        obj.EfekPilihan.Add(delegate()
        {
            Status.sanity = Status.sanity + 50;
            if (Status.sanity >= 100)
            {
                Status.sanity = 100;
            }
            Status.life = Status.life - 25;
            if (Status.life <= 0)
            {
                Status.life = 0;
                Status.characterlife = false;
            }
        });
    }
}

```

Kode Sumber 4.11 Contoh input life dan sanity

4.3.3 Tempat

Di dalam permainan simulasi bertahan hidup di pulau terpencil ini terdapat 7 tempat yang bisa dikunjungi oleh pemain. Pemain dapat mengunjungi home untuk lanjut ke waktu berikutnya atau mengunjungi selain home untuk menerima event. Saat pemain memulai sesi permainan dan masuk event, pemain secara default mulai dari tempat *home*. Bukti bahwa pemain mulai dari home dapat dilihat dari panah berwarna biru diatas tombol home seperti yang terlihat pada gambar 4.15. Sedangkan kode program untuk mengatur pemain mulai dari tempat tinggal home dapat dilihat pada Kode Sumber 4.12 dan 4.13.



Gambar 4.15 Pemain mulai dari home

```
public class Status
{
    public static string TempatSekarang = "Tempat Berada Home";

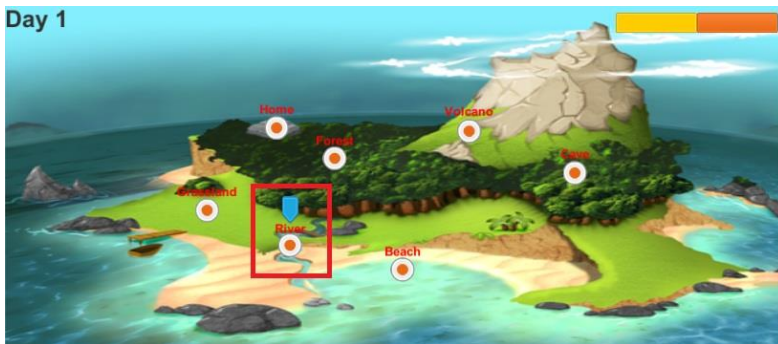
    public static void SetStatusUlang()
    {
        TempatSekarang = "Tempat Berada Home";
    }
}
```

Kode Sumber 4.12 Mengatur tempat berada

```
public class MainMenu : MonoBehaviour
{
    void Start()
    {
        Status.SetStatusUlang();
    }
}
```

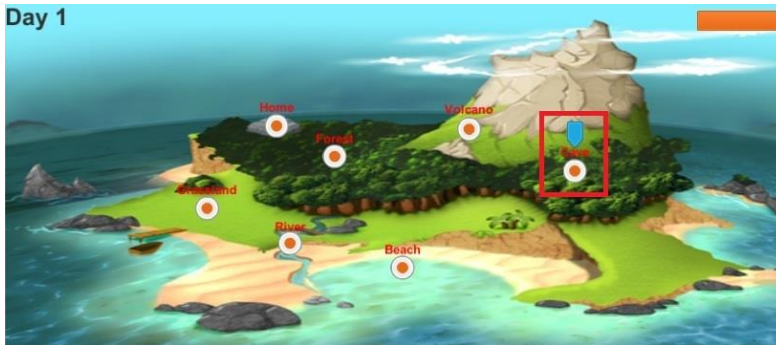
Kode Sumber 4.13 Mengatur tempat berada saat mulai

Panah biru diatas tombol *home* digunakan sebagai penunjuk pemain berada sekarang. Panah biru tersebut akan berpindah ke tempat lain jika kita mengunjungi tempat lain selain kita berada sekarang. Contoh, pada awal pemain memulai permainan pemain berada di *home*. Lalu pemain menekan tombol sungai dan menerima event di sungai. Di waktu berikutnya, panah biru yang sebelumnya berada diatas tombol *home* langsung berpindah ke diatas tombol sungai seperti yang terlihat pada gambar 4.16.



Gambar 4.16 Pemain berada di sungai

Setelah itu, pemain menekan tombol gua dan menerima event di gua. Di waktu berikutnya, panah biru tersebut kembali berpindah tempat. Kali ini panah biru yang sebelumnya berada diatas tombol sungai berpindah ke diatas tombol gua seperti yang terlihat pada gambar 4.17



Gambar 4.17 Pemain berada di gua

Lalu pemain memilih tempat manapun dan masuk ke hari berikutnya. Saat pemain masuk ke hari berikutnya, panah biru tersebut kembali lagi diatas tombol *home* meskipun pemain sebelumnya tidak memilih tombol *home* seperti yang terlihat pada gambar 4.18. Kode program untuk menentukan tempat pemain berada dapat dilihat pada Kode Sumber 4.15 dan 4.16. Sedangkan kode program untuk input memberitahu dimana pemain berada dapat dilihat pada Kode Sumber 4.17.



Gambar 4.18 Pemain berada di home kembali waktu masuk ke hari berikutnya

```

using UnityEngine.UI;

public class Event : MonoBehaviour
{
    private GameObject PenandaHome;
    private GameObject PenandaHutan;
    private GameObject PenandaPadangRumput;
    private GameObject PenandaSungai;
    private GameObject PenandaPantai;
    private GameObject PenandaGunungBerapi;
    private GameObject PenandaGua;
    void Start()
    {
        PenandaHome = this.gameObject.transform.GetChild(19).gameObject;
        PenandaPadangRumput =
this.gameObject.transform.GetChild(21).gameObject;
        PenandaHutan = this.gameObject.transform.GetChild(20).gameObject;
        PenandaSungai =
this.gameObject.transform.GetChild(22).gameObject;
        PenandaPantai = this.gameObject.transform.GetChild(23).gameObject;
        PenandaGunungBerapi =
this.gameObject.transform.GetChild(24).gameObject;
        PenandaGua = this.gameObject.transform.GetChild(25).gameObject;
        PenandaHome.SetActive(true);
        PenandaHutan.SetActive(false);
        PenandaPadangRumput.SetActive(false);
        PenandaSungai.SetActive(false);
        PenandaPantai.SetActive(false);
        PenandaGunungBerapi.SetActive(false);
        PenandaGua.SetActive(false);
    }
}

```

Kode Sumber 4.14 Membaca objek panah biru di masing-masing tempat dan pemasangan awal

```

public class Event : MonoBehaviour
{
    void Start()
    {
        switch(Status.TempatSekarang)
        {
            case "Tempat Berada Home":
                PenandaHome.SetActive(true);
                PenandaHutan.SetActive(false);
                PenandaPadangRumput.SetActive(false);
                PenandaSungai.SetActive(false);
                PenandaPantai.SetActive(false);
                PenandaGunungBerapi.SetActive(false);
                PenandaGua.SetActive(false);
                break;
            case "Tempat Berada Hutan":
                PenandaHome.SetActive(false);
                PenandaHutan.SetActive(true);
                PenandaPadangRumput.SetActive(false);
                PenandaSungai.SetActive(false);
                PenandaPantai.SetActive(false);
                PenandaGunungBerapi.SetActive(false);
                PenandaGua.SetActive(false);
                break;
            case "Tempat Berada Padang Rumput":
                PenandaHome.SetActive(false);
                PenandaHutan.SetActive(false);
                PenandaPadangRumput.SetActive(true);
                PenandaSungai.SetActive(false);
                PenandaPantai.SetActive(false);
                PenandaGunungBerapi.SetActive(false);
                PenandaGua.SetActive(false);
                break;
            case "Tempat Berada Sungai":
                PenandaHome.SetActive(false);
                PenandaHutan.SetActive(false);
                PenandaPadangRumput.SetActive(false);
                PenandaSungai.SetActive(true);
                PenandaPantai.SetActive(false);
                PenandaGunungBerapi.SetActive(false);
                PenandaGua.SetActive(false);
                break;
            case "Tempat Berada Pantai":

```

Kode Sumber 4.15 Menampilkan tempat pemain berada

```

using UnityEngine.UI;

public class Event : MonoBehaviour
{
    public void ButtonHome()
    {
        Status.TempatSekarang = "Tempat Berada Home";
    }
    public void ButtonHutan()
    {
        Status.TempatSekarang = "Tempat Berada Hutan";
    }
    public void ButtonPadangRumput()
    {
        Status.TempatSekarang = "Tempat Berada Padang Rumput";
    }
    public void ButtonSungai()
    {
        Status.TempatSekarang = "Tempat Berada Sungai";
    }
    public void ButtonPantai()
    {
        Status.TempatSekarang = "Tempat Berada Pantai";
    }
    public void ButtonGunungBerapi()
    {
        Status.TempatSekarang = "Tempat Berada Gunung Berapi";
    }
    public void ButtonGua()
    {
        Status.TempatSekarang = "Tempat Berada Gua";
    }
}

public void GantiWaktu()
{
    switch (Status.WaktuHariSekarang)
    {
        case "Sore":
            Status.WaktuHariSekarang = "Pagi";
            Status.day++;
            Status.TempatSekarang = "Tempat Berada Home";
    }
}

```

Kode Sumber 4.16 Input tempat pemain berada dan kembali ke home di hari berikutnya

4.3.4 Waktu

Pada saat pemain memulai permainan, pemain selalu dimulai dari hari ke-1. Lalu setiap harinya, setiap kali pemain masuk ke bagian *event*, pemain diberi 3 bar waktu, yaitu pagi, siang, dan sore. Setelah melewati ke 3 waktu tersebut, pemain masuk ke hari berikutnya dan jumlah hari bertambah. Kode program untuk memasang waktu dan hari dapat dilihat pada Kode Sumber 4.18 dan 4.19.

```
public class Status
{
    public static int day = 1;

    public static string WaktuHariSekarang = "Pagi";

    public static void SetStatusUlang()
    {
        day = 1;

        WaktuHariSekarang = "Pagi";
    }
}
```

Kode Sumber 4.17 Mengatur waktu dan hari

```
public class MainMenu : MonoBehaviour
{
    void Start()
    {
        Status.SetStatusUlang();
    }
}
```

Kode Sumber 4.18 Mengatur waktu dan hari saat mulai permainan

Kode program untuk menampilkan hari di bagian awal dapat dilihat pada Kode Sumber 4.20 dan menampilkan waktu dan hari di bagian event dapat dilihat pada Kode Sumber 4.21.

```
public class Utama : MonoBehaviour
{
    void Start()
    {
        int day = Status.day;
        ThisDay.GetComponent<Text>().text = "Day " + day.ToString();
    }
}
```

Kode Sumber 4.19 Menampilkan hari di bagian event

```
public class Event : MonoBehaviour
{
    private GameObject BarPagi;
    private GameObject BarSiang;
    private GameObject BarSore;

    void Start()
    {
        int day = Status.day;
        GameObject ThisDay = this.gameObject.transform.GetChild(11).gameObject;

        ThisDay.GetComponent<Text>().text = "Day " + day.ToString();
        BarPagi = this.gameObject.transform.GetChild(8).gameObject;
        BarSiang = this.gameObject.transform.GetChild(9).gameObject;
        BarSore = this.gameObject.transform.GetChild(10).gameObject;
        BarPagi.SetActive(true);
        BarSiang.SetActive(true);
        BarSore.SetActive(true);
    }
}
```

Kode Sumber 4.20 Menampilkan waktu dan hari di event

Saat pemain masuk ke bagian event, pemain selalu mulai dari waktu pagi. Saat masuk di pagi hari, pemain bisa melihat ketiga bar waktu (biru, kuning, oranye) tersebut ada di pojok kanan atas tampilan seperti pada Gambar 4.15.

Lalu setelah menyelesaikan event di pagi hari atau langsung lanjut ke siang hari melalui tombol *home*, pemain masuk di siang hari. Saat di siang hari, pemain bisa melihat di pojok kanan atas tampilan bar berwarna biru menghilang dan tinggal 2 bar tersisa (kuning dan oranye) yang ada di tampilan seperti pada Gambar 4.16.

Lalu setelah menyelesaikan event di siang hari atau langsung lanjut ke sore hari melalui tombol *home*, pemain masuk ke sore hari. Saat di sore hari, pemain bisa melihat di pojok kanan atas tampilan bar berwarna biru dan kuning menghilang dan tinggal bar oranye yang ada di tampilan seperti pada Gambar 4.17

Setelah menyelesaikan event di sore hari atau langsung lanjut melalui tombol *home*, pemain masuk ke hari berikutnya. Saat masuk ke hari berikutnya, jumlah hari yang dilalui pemain ditambah 1 dan di pojok kanan atas tampilan, ketiga bar waktu muncul di tampilan.

Kode program untuk pengecekan waktu dapat dilihat pada Kode Sumber 4.22. Sedangkan kode program untuk input mengganti waktu dan hari dapat dilihat pada Kode Sumber 4.23 dan 4.24


```

public class Event : MonoBehaviour
{
    void Start()
    {
        switch (Status.WaktuHariSekarang)
        {
            case "Pagi":
                break;
            case "Siang":
                BarPagi.SetActive(false);
                break;
            case "Sore":
                BarPagi.SetActive(false);
                BarSiang.SetActive(false);
                break;
        }
    }
}

```

Kode Sumber 4.21 Mengecek Waktu

```

public class Event : MonoBehaviour
{
    public void GantiWaktu()
    {
        switch (Status.WaktuHariSekarang)
        {
            case "Pagi":
                Status.WaktuHariSekarang = "Siang";
                break;
            case "Siang":
                Status.WaktuHariSekarang = "Sore";
                break;
            case "Sore":
                Status.WaktuHariSekarang = "Pagi";
                Status.day++;
                break;
        }
    }
}

```

Kode Sumber 4.22 Input mengganti waktu dan hari

4.3.5 Kejadian

Untuk menerima kejadian atau event, di bagian *event* pemain harus memilih salah satu dari 6 tempat selain *home*. Setelah pemain memilih salah satu tempat, pemain menerima *event* dan *event bar* akan muncul. Kode program pengaturan awal *event bar* sebelum pemain memilih ke 6 tempat untuk menerima *event* dapat dilihat pada Kode Sumber 4.24

```
public class Event : MonoBehaviour
{
    private GameObject EventBar;
    private GameObject EventText;
    private GameObject Pilihan1;
    private GameObject Pilihan2;
    private GameObject Pilihan3;
    private GameObject Pilihan4;
    private GameObject TextPilihan1;
    private GameObject TextPilihan2;
    private GameObject TextPilihan3;
    private GameObject TextPilihan4;
    private GameObject NextButton;
    void Start()
    {
        EventBar = this.gameObject.transform.GetChild(26).gameObject;
        EventText = this.gameObject.transform.GetChild(26).GetChild(0).gameObject;
        Pilihan1 = this.gameObject.transform.GetChild(26).GetChild(1).gameObject;
        Pilihan2 = this.gameObject.transform.GetChild(26).GetChild(2).gameObject;
        Pilihan3 = this.gameObject.transform.GetChild(26).GetChild(3).gameObject;
        Pilihan4 = this.gameObject.transform.GetChild(26).GetChild(4).gameObject;
        TextPilihan1 =
this.gameObject.transform.GetChild(26).GetChild(1).GetChild(0).gameObject;
        TextPilihan2 =
this.gameObject.transform.GetChild(26).GetChild(2).GetChild(0).gameObject;
        TextPilihan3 =
this.gameObject.transform.GetChild(26).GetChild(3).GetChild(0).gameObject;
        TextPilihan4 =
this.gameObject.transform.GetChild(26).GetChild(4).GetChild(0).gameObject;
        NextButton = this.gameObject.transform.GetChild(26).GetChild(5).gameObject;
        EventBar.SetActive(false);
    }
}
```

Kode Sumber 4.23 Pengaturan awal event bar

Setelah pemain memilih tempat yang dipilih untuk menerima event, aplikasi mengakses kumpulan-kumpulan event dari tempat yang dipilih. Lalu aplikasi mengambil salah satu event dari kumpulan-kumpulan event dari tempat yang dipilih secara acak. Lalu aplikasi memunculkan event bar yang berisikan teks event dari event yang dipilhkan, beserta pilihan-pilihannya (jika ada), dan tombol ke waktu berikutnya (jika event tidak memiliki pilihan) seperti yang terlihat pada gambar 4.19. Kode program untuk aplikasi mengakses kumpulan-kumpulan event dari salah satu tempat yang dipilih dapat dilihat pada Kode Sumber 4.25, 4.26, dan 4.27.



Gambar 4.19 Menerima event dari tempat yang dipilih

```

public class Event : MonoBehaviour
{
    public void ButtonGua()
    {
        int AngkaAcak = Random.Range(0, 5);
        DaftarEventGua.SetEvent();
        EventSekarang = DaftarEventGua.KumpulanEventGua[AngkaAcak];
        EventText.GetComponent<Text>().text = EventSekarang.EventText;
        if (EventSekarang.StatusPilihan1 == true)
        {
            Pilihan1.SetActive(true);
            TextPilihan1.GetComponent<Text>().text = EventSekarang.Pilihan1;

            Pilihan1.GetComponent<Button>().onClick.AddListener(EventSekarang.EfekPilihan[
0]);
            Pilihan1.GetComponent<Button>().onClick.AddListener(SetelahPilihan1);
        }
        if (EventSekarang.StatusPilihan2 == true)
        {
            Pilihan2.SetActive(true);
            TextPilihan2.GetComponent<Text>().text = EventSekarang.Pilihan2;

            Pilihan2.GetComponent<Button>().onClick.AddListener(EventSekarang.EfekPilihan[
1]);
            Pilihan2.GetComponent<Button>().onClick.AddListener(SetelahPilihan2);
        }
        if (EventSekarang.StatusPilihan3 == true)
        {
            Pilihan3.SetActive(true);
            TextPilihan3.GetComponent<Text>().text = EventSekarang.Pilihan3;

            Pilihan3.GetComponent<Button>().onClick.AddListener(EventSekarang.EfekPilihan[
2]);
            Pilihan3.GetComponent<Button>().onClick.AddListener(SetelahPilihan3);
        }
        if (EventSekarang.StatusPilihan4 == true)
        {
            Pilihan4.SetActive(true);
            TextPilihan4.GetComponent<Text>().text = EventSekarang.Pilihan4;

            Pilihan4.GetComponent<Button>().onClick.AddListener(EventSekarang.EfekPilihan[
3]);
            Pilihan4.GetComponent<Button>().onClick.AddListener(SetelahPilihan4);
        }
    }
}

```

```

        if (EventSekarang.TidakAdaPilihan == true)
        {
            NextButton.SetActive(true);
        }
    }
}

```

Kode Sumber 4.24 Mengakses kumpulan event gua dan menampilkan event yang dipilih

```

public class DaftarEventGua
{
    public static List<EventObject> KumpulanEventGua = new List<EventObject>();
    public static void SetEvent()
    {
        KumpulanEventGua = new List<EventObject>();
        new EventGua1();
        new EventGua2();
        new EventGua3();
        new EventGua4();
    }
}

```

Kode Sumber 4.25 Menyusun kumpulan event-event gua

```

public class MainMenu : MonoBehaviour
{
    // Use this for initialization
    void Start()
    {
        if (DaftarEventHutan.KumpulanEventHutan.Count == 0)
        {
            DaftarEventHutan.SetEvent();
        }
        if (DaftarEventPadangRumput.KumpulanEventPadangRumput.Count == 0)
        {
            DaftarEventPadangRumput.SetEvent();
        }
    }
}

```

```

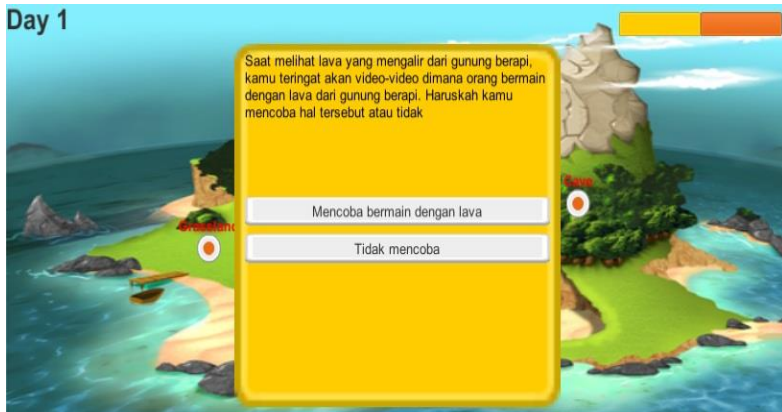
if (DaftarEventSungai.KumpulanEventSungai.Count == 0)
{
    DaftarEventSungai.SetEvent();
}
if (DaftarEventPantai.KumpulanEventPantai.Count == 0)
{
    DaftarEventPantai.SetEvent();
}
if (DaftarEventGunungBerapi.KumpulanEventGunungBerapi.Count == 0)
{
    DaftarEventGunungBerapi.SetEvent();
}
if (DaftarEventGua.KumpulanEventGua.Count == 0)
{
    DaftarEventGua.SetEvent();
}
}
}

```

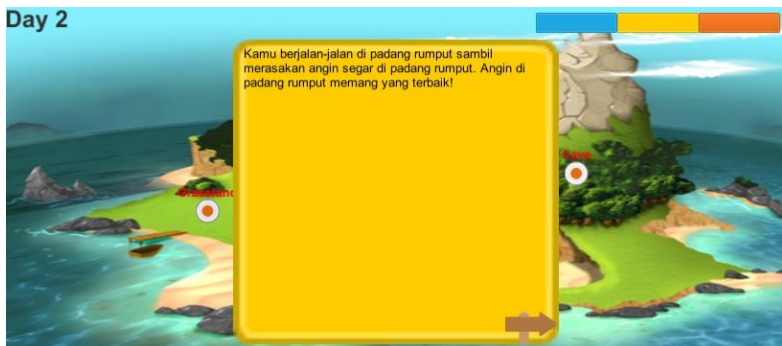
Kode Sumber 4.26 Menyusun dan mengakses kumpulan event dari semua tempat saat memulai permainan

Event yang diterima oleh pemain di permainan simulasi bertahan hidup ini bermacam-macam, dan efek pilihan yang diterima pemain dari pilihan yang dipilih dari suatu *event* juga bermacam-macam.

Pertama, ada event yang memiliki efek pilihan menambah atau mengurangi status yang contohnya seperti pada Gambar 4.20. Kode program untuk contoh event dengan model efek pilihan yang menambah atau mengurangi status dapat dilihat pada Kode Sumber 4.28.



Gambar 4.20 Contoh event menambah atau mengurangi status



Gambar 4.21 Contoh event biasa tanpa pilihan

Kedua, ada event biasa yang tidak memiliki pilihan yang contohnya seperti pada Gambar 4.21. Kode program untuk contoh event dengan model tidak ada pilihan dapat dilihat pada Kode Sumber 4.29.

```

public class EventGunungBerapi2
{
    public EventGunungBerapi2()
    {
        EventObject obj = new EventObject();
        obj.EventText = "Saat melihat lava yang mengalir dari gunung berapi,
        kamu teringat akan video-video dimana orang bermain dengan lava dari
        gunung berapi. Haruskah kamu mencoba hal tersebut atau tidak";
        obj.StatusPilihan1 = true;
        obj.Pilihan1 = "Mencoba bermain dengan lava";
        obj.TextSetelahPilihan1 = "Sanity kamu bertambah, life kamu
        berkurang sedikit";
        obj.StatusPilihan2 = true;
        obj.Pilihan2 = "Tidak mencoba";
        obj.TextSetelahPilihan2 = "Tidak terjadi apa-apa";
        obj.EfekPilihan.Add(delegate()
        {
            Status.sanity = Status.sanity + 50;
            if (Status.sanity >= 100)
            {
                Status.sanity = 100;
            }
            Status.life = Status.life - 25;
            if (Status.life <= 0)
            {
                Status.life = 0;
                Status.characterlife = false;
            }
        });
        obj.EfekPilihan.Add(delegate()
        {
        });
        DaftarEventGunungBerapi.KumpulanEventGunungBerapi.Add(obj);
    }
}

```

Kode Sumber 4.27 Contoh event dengan efek pilihan menambah atau mengurangi status


```

public class EventPadangRumput1
{
    public EventPadangRumput1()
    {
        EventObject obj = new EventObject();
        obj.EventText = "Kamu berjalan-jalan di padang rumput sambil merasakan angin
segar di padang rumput. Angin di padang rumput memang yang terbaik!";
        obj.TidakAdaPilihan = true;
        DaftarEventPadangRumput.KumpulanEventPadangRumput.Add(obj);
    }
}

```

Kode Sumber 4.28 Contoh event biasa tanpa pilihan

Lalu ada event yang memiliki efek pilihan menambahkan barang yang contohnya seperti pada Gambar 4.22. Setelah mendapatkan barang, barang yang didapatkan dari event langsung ditambahkan dan ditampilkan namanya di menu inventory seperti pada gambar 4.23. Kode program untuk contoh event dengan efek pilihan menambahkan barang dapat dilihat pada Kode Sumber 4.30. Sedangkan kode program untuk menampilkan nama barang di inventory dapat dilihat pada Kode Sumber 4.31.



Gambar 4.22 Contoh event mendapatkan barang

```

public class EventHutan1
{
    public EventHutan1()
    {
        EventObject obj = new EventObject();
        obj.EventText = "Saat kamu berkeliling di hutan, kamu menemukan
        kayu dan benang laba-laba yang sangat kuat. Sepertinya kamu dapat
        membuat sesuatu dari benda-benda tersebut. Apa yang akan kamu buat";
        obj.StatusPilihan1 = true;
        obj.Pilihan1 = "Panah";
        obj.TextSetelahPilihan1 = "Kamu mendapatkan Panah";
        obj.StatusPilihan2 = true;
        obj.Pilihan2 = "Alat Pancing";
        obj.TextSetelahPilihan2 = "Kamu mendapatkan Alat pancing";
        obj.EfekPilihan.Add(delegate()
        {
            switch (Barang.panahstatus)
            {
                case false:
                    Barang.panahstatus = true;
                    Barang.ListBarang.Add("Panah");
                    break;
            }
        });
        obj.EfekPilihan.Add(delegate()
        {
            switch (Barang.alatpancingstatus)
            {
                case false:
                    Barang.alatpancingstatus = true;
                    Barang.ListBarang.Add("Alat Pancing");
                    break;
            }
        });
        DaftarEventHutan.KumpulanEventHutan.Add(obj);
    }
}

```

Kode Sumber 4.29 Contoh event mendapatkan barang

```

public class Inventory : MonoBehaviour
{
    void Start()
    {
        int BanyakBarang = Barang.ListBarang.Count;
        Barang.TotalPage = (int)Math.Ceiling((double)BanyakBarang / 12);

        if(Barang.TotalPage == 0){
            Barang.TotalPage = 1;
        }

        for(int i = 0; i<12; i++){
            int IndexBarang = (Barang.CurrentPage - 1) * 12 + i;
            if(IndexBarang < BanyakBarang){
                GameObject NamaBarangdanJumlah =
                this.gameObject.transform.GetChild(1).GetChild(i+4).gameObject;
                if(Barang.ListBarang[IndexBarang].Equals("Makanan"))
                {
                    NamaBarangdanJumlah.GetComponent<Text>().text =
                    Barang.ListBarang[IndexBarang] + " " + Barang.makanan;
                }
                else if(Barang.ListBarang[IndexBarang].Equals("Minuman"))
                {
                    NamaBarangdanJumlah.GetComponent<Text>().text =
                    Barang.ListBarang[IndexBarang] + " " + Barang.minuman;
                }
                else if(Barang.ListBarang[IndexBarang].Equals("Anak Panah"))
                {
                    NamaBarangdanJumlah.GetComponent<Text>().text =
                    Barang.ListBarang[IndexBarang] + " " + Barang.anakpanah;
                }
                else
                {
                    NamaBarangdanJumlah.GetComponent<Text>().text =
                    Barang.ListBarang[IndexBarang];
                }
            }
            else{
                GameObject NamaBarangdanJumlah =
                this.gameObject.transform.GetChild(1).GetChild(i + 4).gameObject;
                NamaBarangdanJumlah.SetActive(false);
            }
        }
    }
}

```

```

        if (Barang.CurrentPage == 1)
        {
            GameObject PreviousPage =
this.gameObject.transform.GetChild(1).GetChild(3).gameObject;
            PreviousPage.SetActive(false);
        }
        if (Barang.CurrentPage == Barang.TotalPage)
        {
            GameObject NextPage =
this.gameObject.transform.GetChild(1).GetChild(2).gameObject;
            NextPage.SetActive(false);
        }

        GameObject PageText =
this.gameObject.transform.GetChild(1).GetChild(16).gameObject;
        PageText.GetComponent<Text>().text = Barang.CurrentPage.ToString() + " / "
+ Barang.TotalPage.ToString();
    }
}

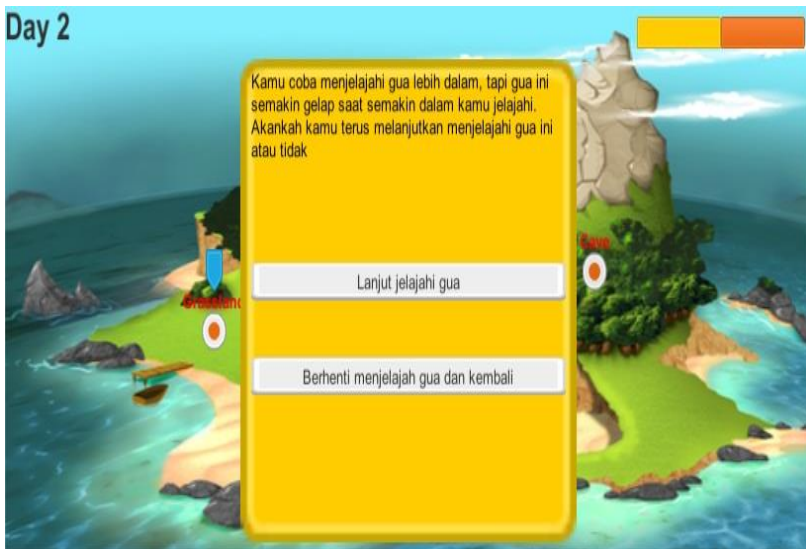
```

Kode Sumber 4.30 Menampilkan barang di menu inventory



Gambar 4.23 Barang di menu inventory setelah mendapatkan panah

Terakhir, ada event yang memerlukan suatu barang untuk bisa mengakses salah satu pilihannya yang contohnya seperti pada Gambar 4.24. Untuk event model ini, Jika pemain memiliki barang yang diperlukan, pemain dapat mengakses dan memilih pilihan tersebut, tapi jika pemain tidak memiliki barang yang diperlukan, pemain tidak bisa mengakses pilihan tersebut, sehingga pemain juga tidak bisa memilih pilihan tersebut. Kode program untuk event yang ada pilihannya memerlukan suatu barang untuk bisa diakses dapat dilihat pada Kode Sumber 4.32.



Gambar 4.24 Contoh event yang memerlukan barang untuk mengakses salah satu pilihannya

```

public class EventGua4
{
    public EventGua4()
    {
        EventObject obj = new EventObject();
        obj.EventText = "Kamu coba menjelajahi gua lebih dalam, tapi gua ini semakin
        gelap saat semakin dalam kamu jelajahi. Akankah kamu terus melanjutkan
        menjelajahi gua ini atau tidak";
        obj.StatusPilihan1 = true;
        obj.Pilihan1 = "Lanjut jelajahi gua";
        obj.TextSetelahPilihan1 = "Life kamu berkurang sedikit";
        if (Barang.oborstatus == true)
        {
            obj.StatusPilihan2 = true;
            obj.Pilihan2 = "Gunakan Obor";
            obj.TextSetelahPilihan2 = "Mendapatkan tali";
        }
        obj.StatusPilihan3 = true;
        obj.Pilihan3 = "Berhenti menjelajah gua dan kembali";
        obj.TextSetelahPilihan3 = "Tidak terjadi apa-apa";
        obj.EfekPilihan.Add(delegate()
        {
            Status.life = Status.life - 10;
            if (Status.life <= 0)
            {
                Status.life = 0;
                Status.characterlife = false;
            }
        });
        obj.EfekPilihan.Add(delegate()
        {
            Barang.talistatus = true;
            Barang.ListBarang.Add("Tali");
        });
        obj.EfekPilihan.Add(delegate()
        {
        });
        DaftarEventGua.KumpulanEventGua.Add(obj);
    }
}

```

Kode Sumber 4.31 Contoh event yang memerlukan barang untuk mengakses salah satu pilihannya

(Halaman Ini Sengaja Dikosongkan)

BAB V

PENGUJIAN DAN EVALUASI

Pada bab ini akan dijelaskan mengenai rangkaian uji coba dan evaluasi yang dilakukan.

5.1 Lingkungan Uji Coba

Lingkungan pengujian sistem pada pengerjaan Tugas Akhir ini dilakukan pada lingkungan dan alat kaka sebagai berikut:

Jenis Perangkat : Xiaomi Redmi Note 3

Processor : Hexa-core Max 1.8 GHz

Memori : 3 GB RAM

Sistem Operasi : Android 5.1.1 Lollipop

5.2 Pengujian Aturan Menang Dan Kalah Dan Faktor-Faktor Bertahan Hidup

Pengujian ini dilakukan untuk mengetahui kesesuaian keluaran dari tiap faktor-faktor bertahan hidup terhadap skenario yang dipersiapkan. Penjabaran skenario dan hasil uji coba yang dilakukan terhadap perangkat lunak yang dibangun dapat dilihat pada Tabel 5.1, 5.2, 5.3, dan 5.4.

Tabel 5.1 Uji coba aturan menang dan kalah

ID	UF-001
Nama	Uji Coba Aturan Menang Dan Kalah

ID	UF-001
Tujuan Uji Coba	Menguji sistem bisa masuk ke scene menang dan scene kalah sesuai aturan yang ditetapkan
Kondisi Awal	Pemain sudah memulai sesi permainan dan berada di bagian awal
Skenario 1	Pemain memilih tidak makan dan minum selama 4 hari berturut-turut
Keluaran yang diharapkan	Pemain masuk ke scene kalah
Hasil uji coba	Berhasil
Kondisi akhir	Sistem masuk ke scene kalah sesuai aturan kalah yang ditetapkan
Skenario 2	Pemain bertahan hidup hingga hari ke-100
Keluaran yang diharapkan	Pemain masuk ke scene menang
Hasil uji coba	Berhasil
Kondisi akhir	Sistem masuk ke scene menang sesuai aturan menang yang ditetapkan

Tabel 5.2 Uji coba status hungry dan thirsty

ID	UF-002
Nama	Uji Coba Finite State Machine Hungry Dan Thirsty
Tujuan Uji Coba	Menguji status hungry dan thirsty sudah sesuai dengan finite state machine yang dibuat atau tidak
Kondisi Awal	Pemain sudah memulai sesi permainan dan berada di bagian awal. Status hungry dan thirsty berada ditingkatan healthy
Skenario 1	Pemain masuk ke bagian makan dan minum. Lalu memilih makan dan minum
Keluaran yang diharapkan	Status hungry dan thirsty pemain tetap di healthy
Hasil uji coba	Berhasil
Kondisi akhir	Status hungry dan thirsty pemain tetap di healthy di hari berikutnya
Skenario 2	Pemain masuk ke bagian makan dan minum. Lalu memilih tidak makan dan minum
Keluaran yang diharapkan	Status hungry dan thirsty pemain berubah menjadi little hungry
Hasil uji coba	Berhasil

ID	UF-002
Kondisi akhir	Status hungry dan thirsty pemain berubah menjadi little hungry di hari berikutnya
Skenario 3	Pemain masuk ke bagian makan dan minum. Lalu memilih makan dan tidak minum
Keluaran yang diharapkan	Status hungry dan thirsty pemain berubah menjadi thirsty
Hasil uji coba	Berhasil
Kondisi akhir	Status hungry dan thirsty pemain berubah menjadi thirsty di hari berikutnya
Skenario 4	Pemain masuk ke bagian makan dan minum. Lalu memilih tidak makan dan tidak minum
Keluaran yang diharapkan	Status hungry dan thirsty pemain berubah menjadi little hungry and thirsty
Hasil uji coba	Berhasil
Kondisi akhir	Status hungry dan thirsty pemain berubah menjadi little hungry and thirsty

Tabel 5.3 Uji coba status life

ID	UF-003
Nama	Uji Coba Finite State Machine Life
Tujuan Uji Coba	Menguji status life sudah sesuai dengan finite state machine yang dibuat atau tidak
Kondisi Awal	Pemain sudah memulai sesi permainan dan berada di bagian event. Status life berada ditingkatan healthy
Skenario 1	Pemain menerima event dan memilih pilihan yang memberikan efek yang mengurangi life total sebesar 50 hingga ke hari berikutnya
Keluaran yang diharapkan	Status life pemain berubah menjadi weak
Hasil uji coba	Berhasil
Kondisi akhir	Status life pemain berubah menjadi weak di hari berikutnya

Tabel 5.4 Uji coba status sanity

ID	UF-004
Nama	Uji Coba Finite State Machine Sanity

ID	UF-004
Tujuan Uji Coba	Menguji status sanity sudah sesuai dengan finite state machine yang dibuat atau tidak
Kondisi Awal	Pemain sudah memulai sesi permainan dan berada di bagian event. Status sanity berada ditingkatan sane
Skenario 1	Pemain menerima event dan memilih pilihan yang memberikan efek yang mengurangi sanity total sebesar 50 hingga ke hari berikutnya
Keluaran yang diharapkan	Status sanity pemain berubah menjadi crazy
Hasil uji coba	Berhasil
Kondisi akhir	Status sanity pemain berubah menjadi crazy di hari berikutnya

Tabel 5.5 Uji coba event

ID	UF-003
Nama	Uji Coba Event
Tujuan Uji Coba	Menguji sistem bisa memberikan event kepada pemain

ID	UF-003
Kondisi Awal	Pemain sudah memulai sesi permainan dan berada di bagian event di waktu pagi hari
Skenario 1	Pemain memilih salah satu tempat selain home
Keluaran yang diharapkan	Pemain menerima event. Lalu pemain masuk ke waktu berikutnya setelah menyelesaikan event, tempat berada pemain berada di tempat yang dipilih pemain sebelumnya, dan pemain berada di waktu siang hari
Hasil uji coba	Berhasil
Kondisi akhir	Sistem memberikan pemain sebuah event. Begitu masuk ke waktu berikutnya setelah pemain menyelesaikan event, tempat berada pemain sekarang adalah di tempat yang dipilih pemain sebelumnya, dan bar waktu pagi hilang
Skenario 2	Pemain memilih tempat home 3 kali
Keluaran yang diharapkan	Pemain masuk ke hari berikutnya
Hasil uji coba	Berhasil
Kondisi akhir	Pemain berada di hari berikutnya

Tabel 5.6 Menemukan keberadaan faktor-faktor bertahan hidup

NO.	KEBERADAAN FAKTOR-FAKTOR BERTAHAN HIDUP	PADA SKENARIO
1	Aturan menang dan kalah	Skenario 1
2	Status	Skenario 2
3	Tempat	Skenario 3
4	Waktu	Skenario 3
5	Kejadian atau event	Skenario 3

Tabel 5.7 Menemukan implementasi finite state machine di dalam permainan

NO.	IMPLEMENTASI FINITE STATE MACHINE	PADA SKENARIO
1	Hungry dan Thirsty	Skenario 1
2	Life	Skenario 2
3	Sanity	Skenario 3
4	Tempat	Skenario 3

Pada Tabel 5.6 di atas dapat disimpulkan bahwa keberadaan implementasi faktor-faktor bertahan hidup sudah ada pada skenario di dalam aplikasi. Sedangkan pada tabel 5.7 diatas

dapat disimpulkan bahwa finite state machine yang dirancang sudah diimplementasikan dan sudah ada pada skenario di dalam aplikasi.

5.3 Pengujian Pengguna

Evaluasi pengujian pengguna dilakukan dengan menyiapkan sejumlah pertanyaan kuesioner dan melakukan pengambilan data dari responden menggunakan kuesioner tersebut. Penghitungan data kuesioner dilakukan untuk mendapatkan tingkat keberhasilan implementasi faktor-faktor bertahan hidup di skenario dalam perangkat lunak.

Pertanyaan-pertanyaan yang terdapat pada perangkat lunak dijabarkan sebagai berikut:

Tabel 5.8 Pertanyaan-pertanyaan kuesioner

NO.	PARAMETER
	Parameter Antarmuka
1	Mudah dipahami
2	Responsif
3	Kesesuaian kebutuhan fungsional
4	Tampilan Menarik
	Parameter Faktor-Faktor Bertahan Hidup
	Status

NO.	PARAMETER
5	Status mempengaruhi faktor bertahan hidup anda di dalam permainan
6	Status hungry dan thirsty yang dimiliki mempengaruhi tindakan yang dipilih anda saat makan dan minum?
7	Berapa kali anda makan dan/atau minum?
8	Status life yang dimiliki mempengaruhi tindakan yang dipilih anda saat memilih pilihan saat event?
9	Status sanity yang dimiliki mempengaruhi tindakan yang dipilih anda saat memilih pilihan saat event?
	Waktu
10	Waktu mempengaruhi faktor bertahan hidup anda di dalam permainan?
11	Waktu mempengaruhi tempat yang dipilih anda saat di bagian event?
	Tempat
12	Tempat mempengaruhi faktor bertahan hidup anda di dalam permainan?
13	Anda memiliki tempat tertentu yang sering dikunjungi?
14	Tempat apa yang sering dikunjungi?

NO.	PARAMETER
15	Anda telah mengunjungi semua tempat di dalam permainan?
	Kejadian atau Event
16	Event mempengaruhi faktor bertahan hidup anda di dalam permainan?
17	Ada memiliki event yang anda sukai?
18	Ada memiliki event yang anda benci dan ingin sekali dihindari?
19	Anda mencoba semua pilihan di dalam event?
	Menang dan Kalah
20	Apakah anda berhasil memenangkan permainan?
21	Jika tidak, berapa hari anda berhasil bertahan hidup di permainan?

5.4 Evaluasi

Subbab ini membahas mengenai evaluasi terhadap pengujian-pengujian yang telah dilakukan. Dalam hal ini, sesuai dengan kuesioner yang dilampirkan. Jumlah responden yang mengisi kuesioner adalah 33 orang. Dari pengujian pengguna diketahui bahwa:

Tabel 5.9 Rekapitulasi jawaban kuesioner berdasar pertanyaan

NO.	PARAMETER				
	Parameter Antarmuka	SS	S	TS	STS
1	Mudah dipahami	2	18	13	0
2	Responsif	6	19	7	1
3	Kesesuaian kebutuhan fungsional	4	22	6	1
4	Tampilan Menarik	4	12	15	2
	Parameter Faktor-Faktor Bertahan Hidup	SS	S	TS	STS
	Status	26	89	16	1
5	Status mempengaruhi faktor bertahan hidup anda di dalam permainan	6	24	3	0
6	Status hungry dan thirsty yang dimiliki mempengaruhi tindakan yang dipilih anda saat makan dan minum?	14	19	0	0

NO.	PARAMETER				
7	Status life yang dimiliki mempengaruhi tindakan yang dipilih anda saat memilih pilihan saat event?	4	23	6	0
8	Status sanity yang dimiliki mempengaruhi tindakan yang dipilih anda saat memilih pilihan saat event?	2	23	7	1
	Waktu	6	37	21	2
9	Waktu mempengaruhi faktor bertahan hidup anda di dalam permainan	3	19	10	1
10	Waktu mempengaruhi tempat yang dipilih anda saat di bagian event?	3	18	11	1
	Tempat	26	53	17	3
11	Tempat mempengaruhi faktor bertahan hidup anda di dalam permainan	9	20	3	1

NO.	PARAMETER				
12	Anda memiliki tempat tertentu yang sering dikunjungi?	8	19	5	1
13	Anda telah mengunjungi semua tempat di dalam permainan?	9	14	9	1
	Kejadian	31	60	41	0
14	Event mempengaruhi faktor bertahan hidup anda di dalam permainan	14	17	2	0
15	Ada event yang anda sukai?	5	13	15	0
16	Ada event yang anda benci dan ingin sekali dihindari?	5	14	14	0
17	Anda mencoba semua pilihan di dalam event?	7	16	10	0

Keterangan:

SS: Sangat Setuju, S: Setuju, B: Biasa, TS: Tidak Setuju,

STS: Sangat Tidak Setuju.

Tabel 5.10 Rekapitulasi jawaban kuesioner berdasar nilai

NILAI	JUMLAH	
Sangat setuju	105	415
Setuju	310	
Tidak setuju	136	146
Sangat tidak setuju	10	

Dari data yang terdapat pada tabel tersebut, dapat diketahui bahwa faktor-faktor bertahan hidup yang diimplemetasikan sangat mempengaruhi tindakan-tindakan yang diambil oleh pemain karena data jawaban dari responden yang menjawab ke nilai setuju dan sangat setuju mencapai 73.98% dengan 87.12% responden setuju atau sangat setuju untuk status, 79.8% responden setuju atau sangat setuju untuk tempat, 65.15% responden setuju atau sangat setuju untuk waktu, dan 68.94% setuju atau sangat setuju untuk kejadian.

Dapat dilihat dari hasil diatas bahwa faktor-faktor bertahan hidup yang paling berpengaruh dari tinggi ke rendah adalah status, tempat, kejadian, dan terakhir waktu.

Tabel 5.11 Rekapitulasi jawaban tempat yang sering dikunjungi

TEMPAT YANG SERING DIKUNJUNGI	JUMLAH
Gua	1
Gunung Berapi	3
Hutan	14
Padang Rumput	2
Pantai	7
Sungai	6

Dari data yang terdapat pada tabel tersebut, dapat diketahui bahwa tempat yang paling sering dikunjungi oleh responden di dalam permainan adalah hutan dengan 42.42% dari jumlah responden. Tempat yang paling sering dikunjungi oleh responden berikutnya adalah pantai dengan 21.21% dari jumlah responden, dan sungai dengan 18.18% dari jumlah responden.

Tabel 5.12 Rekapitulasi jawaban waktu makan dan minum

WAKTU MAKAN DAN MINUM	JUMLAH
Sehari 1x	18
2 hari 1x	11
3 hari 1x	4

Dari data yang terdapat pada tabel tersebut, dapat diketahui bahwa kebanyakan responden makan dan minum sehari 1 kali di

dalam permainan. Hal itu dibuktikan dengan 54.55% dari jumlah responden menjawab 1 hari 1 kali. Sedangkan 33.33% dari jumlah responden menjawab 2 hari 1 kali dan 12.12% dari jumlah responden menjawab 3 hari 1 kali.

Tabel 5.13 Rekapitulasi jawaban memenangkan permainan

MEMENANGKAN PERMAINAN	JUMLAH
Ya	5
Tidak	28

Tabel 5.14 Rekapitulasi jawaban jumlah hari bertahan hidup jika tidak memenangkan permainan

JUMLAH HARI BERTAHAN HIDUP	JUMLAH
<10 hari	21
10-30 hari	7
>30 hari	0

Dari data yang terdapat pada tabel tersebut, dapat diketahui bahwa 15.15% dari jumlah responden berhasil bertahan hidup selama 100 hari dan memenangkan permainan. Sedangkan 84.85% dari jumlah responden tidak berhasil memenangkan permainan. Untuk responden yang tidak memenangkan permainan, 75% cuma berhasil bertahan hidup kurang dari 10 hari, 25% berhasil bertahan hidup selama 10 sampai 30 hari, dan 0% berhasil bertahan hidup selama lebih dari 30 hari.

(Halaman Ini Sengaja Dikosongkan)

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari tujuan pembuatan perangkat lunak dan hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, terdapat pula saran yang ditujukan untuk pengembangan perangkat lunak lebih lanjut.

6.1. Kesimpulan

Setelah proses pengerjaan tugas akhir, mulai dari tahap analisis, desain, hingga implementasi, dilakukan 2 uji coba. Pertama uji aplikasi untuk menemukan keberadaan faktor-faktor bertahan hidup yang diimplementasikan, apakah sudah ada di dalam aplikasi. Dan yang kedua adalah pengambilan kuesioner dari pengguna untuk mengetahui pengaruh faktor-faktor bertahan hidup terhadap tindakan yang diambil pemain.

Dari hasil uji coba pertama dan kedua pada Bab V, dapat ditarik kesimpulan bahwa faktor-faktor bertahan hidup yang diimplementasikan pada aplikasi permainan simulasi bertahan hidup di pulau terpencil ini sangat mempengaruhi tindakan-tindakan yang diambil oleh pemain sebagaimana dari hasil analisis data kuesioner dari pengguna yang menunjukkan bahwa 73.98% menjawab ke nilai setuju dan sangat setuju dengan 87.12% responden setuju atau sangat setuju untuk status, 79.8% responden setuju atau sangat setuju untuk tempat, 65.15% responden setuju atau sangat setuju untuk waktu, dan 68.94% setuju atau sangat setuju untuk kejadian.

6.2. Saran

Berikut merupakan beberapa saran dari responden untuk pengembangan permainan ini di masa yang akan datang, berdasarkan pada hasil perancangan, implementasi dan evaluasi uji coba yang telah dilakukan.

- Bisa ditambahkan goal-goal tertentu, misal membangun tempat berteduh, bertualang, mencari benda tertentu, dan sebagainya.
- Penambahan asset gambar dan music.
- Status sebaiknya diperlihatkan juga saat event sedang berlangsung.
- Lokasi diberi sistem *unlockable* supaya ada keinginan untuk terus main.
- Ditambah aset-aset pendukung lain agar *game* lebih menyenangkan dimainkan.
- UI diperbaiki dan diperbagus, tutorial lebih diperjelas agar pengguna paham dengan maksud permainan.

Sedangkan berikut adalah beberapa saran dari penulis untuk pengembangan permainan ini di masa yang akan datang.

- Ditambahkan aturan menang selain harus bertahan hidup selama 100 hari.
- Ditambahkan kondisi dimana status life dan sanity akan berkurang jika status hungry di level hungry atau dibawahnya, atau/dan status thirsty di level very thirsty atau dibawahnya

DAFTAR PUSTAKA

- [1] “60 Seconds!,” (2014). [Online]. Available: <http://robotgentleman.com/60seconds/>. [Accessed 26 July 2017].
- [2] “60 Seconds – Review,” (2015). [Online]. Available: <http://ramblingfoxreviews.blogspot.co.id/2015/08/60-seconds-review.html/>. [Accessed 26 July 2017]
- [3] Unity, "Game engine, tools and multi platfrom," Unity, [Online]. Available: <http://unity3d.com/unity>. [Accessed 20 December 2016].
- [4] Jones, Ken. 1995. Simulations: A Handbook for Teachers and Trainers.
- [5] Saunders, Danny; Severn, Jacqui. 1999, Simulation and Games for Strategy and Policy Planning.
- [6] Rollings, Andrew; Adams, Ernest. 2006. Fundamentals of Game Design.
- [7] Rollings, Andrew; Adams, Ernest. 2006. Andrew Rollings and Ernest Adams on Game Design.
- [8] Wright, David R. 2005. Finite State Machines
- [9] “Android SDK,” 2017. [Online]. Available: <https://www.techopedia.com/definition/4220/android-sdk/>. [Accessed 7 Juny 2017]

(Halaman Ini Sengaja Dikosongkan)

BIODATA PENULIS



Muhammad Wildianurahman, lahir di Surabaya pada tanggal 29 Maret 1994. Salah satu mahasiswa di S1 Teknik Informatika ITS angkatan tahun 2012. Penulis mengambil bidang minat Interaksi Grafika & Seni (IGS). Penulis memiliki hobi menulis, mendengarkan music, dan bermain game. Penulis pernah mengikuti seminar Schematics 2017 dan seminar Pycon 2017. Selain itu, penulis pernah menjadi juara 2 Maimai MURASAKi competition kategori tag open untuk regional Surabaya. Saat ini Penulis bergabung di beberapa komunitas game seperti STARSLIDER, BEMANI SBY X MLG, dan BEMANI Indonesia. Saat ini penulis sedang tertarik memainkan game rhythm, terutama game rhythm berjudul *Sound Voltex IV Heavenly Haven*, dan sedang berusaha untuk mendapatkan skill level ∞ di *Sound Voltex IV Heavenly Haven* di tahun 2018 ini. Komunikasi dengan penulis dapat menghubungi email: akumunoshinigami@gmail.com